Fishery Data Series No. 08-34

Sonar Estimation of Fall Chum Salmon Abundance in the Sheenjek River, 2004

by

Roger Dunbar

June 2008

Alaska Department of Fish and Game Divisions of Sport Fish and Commercial Fisheries

Symbols and Abbreviations

The following symbols and abbreviations, and others approved for the Système International d'Unités (SI), are used without definition in the following reports by the Divisions of Sport Fish and of Commercial Fisheries: Fishery Manuscripts, Fishery Data Series Reports, Fishery Management Reports, and Special Publications. All others, including deviations from definitions listed below, are noted in the text at first mention, as well as in the titles or footnotes of tables, and in figure or figure captions.

Weights and measures (metric)

centimeter	cm
deciliter	dL
gram	g
hectare	ha
kilogram	kg
kilometer	km
liter	L
meter	m
milliliter	mL
millimeter	mm

Weights and measures (English)

cubic feet per second	ft^3/s
foot	ft
gallon	gal
inch	in
mile	mi
nautical mile	nmi
ounce	oz
pound	lb
quart	qt
yard	yd

Time and temperature

day	d
degrees Celsius	°C
degrees Fahrenheit	°F
degrees kelvin	K
hour	h
minute	min
second	s

Physics and chemistry

all atomic symbols	
alternating current	AC
ampere	A
calorie	cal
direct current	DC
hertz	Hz
horsepower	hp
hydrogen ion activity (negative log of)	pH
parts per million	ppm
parts per thousand	ppt, ‰
volts	V
watts	W

General

Alaska Administrative Code	AAC
all commonly accepted abbreviations	e.g., Mr., Mrs., AM, PM, etc.
all commonly accepted professional titles	e.g., Dr., Ph.D., R.N., etc.
at	@
compass directions:	
east	E
north	N
south	S
west	W
copyright	©
corporate suffixes:	
Company	Co.
Corporation	Corp.
Incorporated	Inc.
Limited	Ltd.
District of Columbia	D.C.
et alii (and others)	et al.
et cetera (and so forth)	etc.
exempli gratia (for example)	e.g.
Federal Information Code	FIC
id est (that is)	i.e.
latitude or longitude	lat. or long.
monetary symbols (U.S.)	\$, ¢
months (tables and figures): first three letters	Jan,...,Dec
registered trademark	®
trademark	™
United States (adjective)	U.S.
United States of America (noun)	USA
U.S.C.	United States Code
U.S. state	use two-letter abbreviations (e.g., AK, WA)

Measures (fisheries)

fork length	FL
mideye-to-fork	MEF
mideye-to-tail-fork	METF
standard length	SL
total length	TL

Mathematics, statistics

all standard mathematical signs, symbols and abbreviations	
alternate hypothesis	H_A
base of natural logarithm	*e*
catch per unit effort	CPUE
coefficient of variation	CV
common test statistics	(F, t, χ^2, etc.)
confidence interval	CI
correlation coefficient (multiple)	R
correlation coefficient (simple)	r
covariance	cov
degree (angular)	°
degrees of freedom	df
expected value	*E*
greater than	>
greater than or equal to	≥
harvest per unit effort	HPUE
less than	<
less than or equal to	≤
logarithm (natural)	ln
logarithm (base 10)	log
logarithm (specify base)	log_{2}, etc.
minute (angular)	'
not significant	NS
null hypothesis	H_O
percent	%
probability	P
probability of a type I error (rejection of the null hypothesis when true)	α
probability of a type II error (acceptance of the null hypothesis when false)	β
second (angular)	"
standard deviation	SD
standard error	SE
variance	
population	Var
sample	var

FISHERY DATA SERIES NO. 08-34

SONAR ESTIMATION OF FALL CHUM SALMON ABUNDANCE IN THE SHEENJEK RIVER, 2004

by
Roger Dunbar
Alaska Department of Fish and Game, Division of Commercial Fisheries, Fairbanks

Alaska Department of Fish and Game
Division of Sport Fish, Research and Technical Services
333 Raspberry Road, Anchorage, Alaska, 99518

June 2008

This investigation was partially financed by the U.S./Canada Salmon Research Cooperative under NOAA Cooperative Agreement Award Number NA04NMF4380264.

The Division of Sport Fish Fishery Data Series was established in 1987 for the publication of technically oriented results for a single project or group of closely related projects. Since 2004, the Division of Commercial Fisheries has also used the Fishery Data Series. Fishery Data Series reports are intended for fishery and other technical professionals. Fishery Data Series reports are available through the Alaska State Library and on the Internet: http://www.sf.adfg.state.ak.us/statewide/divreports/html/intersearch.cfm. This publication has undergone editorial and peer review.

Roger Dunbar,
Alaska Department of Fish and Game, Division of Commercial Fisheries,
1300 College Rd., Fairbanks, AK 99701, USA

This document should be cited as:
Dunbar, R. D. 2008. Sonar estimation of fall chum salmon abundance in the Sheenjek River, 2004. Alaska Department of Fish and Game, Fishery Data Series No. 08-34, Anchorage.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

ABSTRACT

Hydroacoustic Technology, Incorporated (HTI) fixed-location, split-beam, sonar was used to estimate chum salmon *Oncorhynchus keta* escapement in the Sheenjek River August 8 to September 25, 2004. The sonar-estimated escapement was 37,878 chum salmon, 24% below the low end of the Sheenjek River biological escapement goal (BEG) of 50,000–104,000 chum salmon. Median passage was observed on September 13; peak single day passage was September 21 when 3,124 fish were estimated passed the sonar site. A diel migration pattern showed most chum salmon passed the sonar site during periods of darkness or suppressed light. Range of ensonification was considered adequate for most fish which passed near shore. However, the passage estimate should be considered conservative since it does not include fish migrating beyond the counting range (including along the left bank; left and right bank refers to the bank on the left or right side of the river when looking downstream), fish present before sonar equipment was in operation, or fish passing after counting ceased. Only 107 vertebrae samples for age determination were collected because of low salmon passage. Analysis of vertebrae collections showed age-4 fish dominated at 62%, age-5 fish represented 25%, age-3 fish 12%, and age-6 fish about 2% of all fish sampled. Male chum salmon comprised 62% of the sample and 38% were female.

A new multi-beam, dual frequency identification sonar (DIDSON™) developed by Applied Physics Laboratory was tested side-by-side with the currently used HTI sonar. The DIDSON™ system was used to estimate chum salmon passage in the Sheenjek River from August 18 through September 24, 2004. Complete analysis of collected data will be published in a future report.

Key words: Chum salmon, *Oncorhynchus keta,* sonar, hydroacoustics, escapement, enumeration, Yukon River, Porcupine River, Sheenjek River.

INTRODUCTION

Five species of anadromous Pacific salmon *Oncorhynchus* are found in the Yukon River drainage. However, chum salmon *O. keta* are the most abundant and occur in genetically distinct summer and fall runs (Wilmot et al. 1992; Seeb et al. 1995). Fall chum salmon are larger, spawn later, and are less abundant than summer chum salmon. Spawning occurs in upper portions of the drainage in spring fed streams, usually remaining ice-free during the winter (Buklis and Barton 1984). Major fall chum salmon spawning areas occur within the Tanana, Chandalar, and Porcupine river systems, as well as portions of the upper Yukon River in Canada (Figure 1). The Sheenjek River (66° 47.02 N 144° 27.82 W) is one of the most important producers of fall chum salmon in the Yukon River drainage. Located above the Arctic Circle, it heads in glacial ice fields of the Romanzof Mountains, a northern extension of the Brooks Range, and flows southward approximately 400 km to its terminus on the Porcupine River (Figure 2).

INRIVER FISHERIES

Fall chum salmon are harvested for commercial and subsistence uses. Commercial harvest is permitted along the entire Yukon River in Alaska and in the lower portion of the Tanana River. No commercial harvest is permitted in any other tributaries of the drainage including the Koyukuk and Porcupine river systems. Although commercial harvest occurs in the Canadian portion of the Yukon River near Dawson, most fish are taken commercially in the lower river, downstream of the village of Anvik. Subsistence use of fall chum salmon is greatest throughout the upper river drainage, upstream of the village of Koyukuk.

Although the Alaskan commercial fishery for Yukon River fall chum salmon developed in the early 1960s, annual harvests remained relatively low through the early to mid 1970s. Estimated total inriver utilization (U.S. and Canada commercial and subsistence) of Yukon River fall chum salmon was below 300,000 fish per year before the mid 1970s (JTC 2005). Inriver commercial fisheries became more fully developed during the late 1970s and early 1980s, total utilization averaged 535,826 fish from 1979–1983 (Table 1). Harvest peaked in 1979 at 615,377 and in 1981 at 677,257 fish. Since the mid 1980s, management strategies have been implemented to reduce commercial exploitation on fall chum stocks to improve low escapements observed throughout the drainage during the early 1980s. In 1987, the commercial fall chum fishery was completely closed in the Alaskan portion of the drainage. In 1992, commercial fishing in Alaska was restricted to a portion of the Tanana River during the fall season. In addition to a commercial fishery closure, 1993 marked the first year in state history that Alaska Department of Fish and Game (ADF&G) instituted a total closure of subsistence fishing in the Yukon River. The closure was in effect during the latter portion of the fall season in response to the extremely weak fall chum salmon run.

Yukon River fall chum salmon runs improved somewhat from 1994 through 1996. In 1994, limited commercial fishing was permitted in the Alaskan portion of the upper Yukon River, and in the Tanana River. Commercial fishing was permitted in all districts throughout the Alaska portion of the drainage in 1995. In 1996, limited commercial fishing was only permitted in selected districts of the mainstem Yukon River, no commercial fishing was permitted in the Tanana River. Poor salmon runs to Western Alaska from 1997 to 2003 resulted in partial or total closures to commercial and subsistence fishing in Alaskan and Canadian portions of the drainage. Commercial fishing was only permitted in the Tanana River and Canada in 1997. A total commercial fishery closure and limited subsistence fishing was required in 1998. Limited commercial harvest was permitted in 1999, and a total commercial fishery closure and severe subsistence fishing restrictions were required in 2000, 2001, and 2002. Limited commercial fishing for fall chum was allowed in 2003 and 2004. Subsistence harvest of fall chum in 2003 was also limited while the subsistence harvest in 2004 was unrestricted except within the Canadian portion of the Porcupine River.

ESCAPEMENT ASSESSMENT

During the period of 1960–1980, only some segments of Yukon River fall chum salmon runs were estimated from mark–recapture studies (Buklis and Barton 1984). Excluding these tagging studies, and apart from aerial assessment of selected tributaries since the early 1970s, comprehensive escapement estimation studies were sporadic and limited to only 2 streams, the Delta River (Tanana River drainage) and Fishing Branch River (Porcupine River drainage). In the early 1980s, comprehensive escapement assessment studies intensified on major spawning tributaries throughout the drainage.

The Sheenjek River is one of the most intensely monitored fall chum salmon spawning streams in Yukon River drainage. Escapement observations date back to 1960 when USFWS reported chum salmon spawning in September. From 1974 to 1981, escapement observations in the Sheenjek River were limited to aerial surveys flown in late September and early October (Barton 1984). Subsequent to 1980, escapements were monitored annually using fixed-location, single-beam, side-looking sonar systems (Dunbar 2004). However, an early segment of the fall chum salmon run was not included by sonar counting operations from 1981 through 1990 because late project startups centered around August 25. By comparison, average startup during the period of 1991–2003 was August 8, more than 2 weeks earlier than previous years. The sonar-estimated escapements for the years of 1986–1990 were subsequently expanded to include fish passing before sonar operations (Barton 1995). Termination of sonar counting was consistent during the period of 1981–2003, averaging September 25, except in 2000 when the project was terminated early because of extremely low water (Barton 2002).

The Sheenjek River sonar project has estimated fall chum salmon escapement since 1981 and has undergone a number of changes in recent years. The project originally operated Bendix[1] single-beam sonar equipment and, although the Bendix sonar functioned well, the manufacturer ceased production in the mid 1990s and no longer supports the system. In 2000, ADF&G purchased an HTI model 241 split-beam digital echosounder system for use on the Sheenjek River to continue providing the best possible data to fishery managers. In 2000 and 2002, the new system was deployed alongside the existing single-beam sonar and produced results comparable to the Bendix equipment (Dunbar 2004). In 2003 and 2004 the split-beam sonar system was used exclusively to enumerate chum salmon in the Sheenjek River.

Annual escapement estimates averaged 100,533 spawners for the period 1991–2000 and approximately 34,787 spawners for the most recent 5-year period of 1999–2003 (Table 2). From 1992 to 2000, the Sheenjek River minimum biological escapement goal (BEG) was 64,000 fall chum salmon, based upon 1974–1990 aerial indices and hydroacoustic assessment (Buklis 1993). In 2001, ADF&G completed a review of the escapement goal for Yukon River fall chum stocks of which the Sheenjek River assessment is a component. Based on this review of long term escapement, catch, and age composition data, the BEG for the Sheenjek River was set at a range of 50,000–104,000 fall chum salmon (Eggers 2001).

STUDY AREA

The sonar project site is located approximately 10 km upstream from the mouth of the Sheenjek River. Although created by glaciers, the Sheenjek River has numerous clearwater tributaries. Water clarity in the lower river is somewhat unpredictable, but is generally clearest during periods of low water. The water level normally begins to drop in late August and September. Upwelling ground water composes a significant proportion of the river flow volume, especially in winter. It is in these spring areas that fall chum salmon spawn, particularly within the lower 160 km.

Historically, because of unfavorable conditions for transducer placement on the left bank[2], only the right bank of the Sheenjek River has been used to estimate fish passage. Drift gillnet studies in the early 1980s suggested that distribution of the upstream migrant chum salmon was primarily concentrated on the right bank of the river at the sonar site, with only a small but unknown proportion passing on the left bank (Barton 1985). In 2002, ADF&G began testing a new Dual Frequency Identification Sonar (DIDSON™) for counting salmon in small rivers. Based on the results of these tests, which showed this equipment to be easier to use, more accurate, and capable of operating with substrate profiles that are unacceptable for split-beam systems (Maxwell and Gove 2004), the Sheenjek River was selected as an ideal candidate for this system. In an effort to estimate the proportion of fish passing on the left bank, a DIDSON™ was deployed there in 2003. Results indicated that approximately 33% of the fish were migrating up the left bank. Due to large numbers of fish observed on the left bank, ADF&G anticipates operating DIDSON™ on both banks in the future. In 2004, the DIDSON™ was tested side-by-side

---

[1] Product names used in this report are included for scientific completeness, but do not constitute a product endorsement.
[2] Left and right bank refers to the bank on the left or right side of the river when looking downstream.

with the split-beam sonar on the right bank to examine differences in the estimates produced by the 2 types of sonar and whether escapement goals will need to be reevaluated.

OBJECTIVES

Goals for the 2004 Sheenjek River fall chum salmon study were to estimate the timing and magnitude of adult salmon escapement, characterize age and sex composition, and to deploy, test, and compare chum salmon passage estimates of the new DIDSON™ to those of the split-beam sonar system. To accomplish these tasks, these specific objectives were identified:

- Estimate timing and magnitude of chum salmon escapement using fixed-location, split-beam, side-looking hydroacoustic techniques.
- Estimate age and sex composition of the spawning chum salmon population from a minimum of 30–35 vertebrae samples per week up to 180 for the season, such that simultaneous 95% confidence intervals of age composition are no wider than 0.20 (α=0.05 and d=0.10).
- Monitor selected climatological and hydrologic parameters daily at the project site for use as baseline data.
- Deploy and operate the DIDSON™ side by side with the split-beam system on the right bank, and compare the DIDSON™ passage estimates with the split-beam system.

METHODS

HYDROACOUSTIC EQUIPMENT

A fixed-location, split-beam, fisheries hydroacoustic system developed by HTI was used to estimate chum salmon abundance in the Sheenjek River in 2004. Fish passage was monitored with a model 241 digital echo sounder (Appendix A1) and a 2° by 10° 200 kHz split-beam transducer deployed from a right-bank point bar at the historic sonar site (Figures 3 and 4). The transducer was attached to an HTI model 662H dual-axis rotator, with an HTI model 660 remote controller to facilitate aiming. The HTI system is capable of distinguishing upstream fish from downstream fish and debris, determining fish velocity, discriminating between random reverberation and fish targets, and providing a less biased estimate of target strength (HTI 2000).

The HTI digital echo sounder is a state-of-the-art system designed for fisheries research. Accurate time-varied gains (TVG's) and stable transmit and receive sensitivities are possible. Short pulse widths can be used to improve resolution between targets. A Digital Echo Processor (DEP) is integrated into the system. A laptop computer paired with the sounder provides access to all DEP settings and permits saving settings for future use. An oscilloscope can be linked to the sounder for diagnostic use, such as in-situ system calibration or transducer aiming. After all parameters are determined for data acquisition, the system operates 24 hours a day. Files are created by the DEP and edited by the field crew to produce an estimate of fish passage. The crew of 3 technicians monitors the sonar and interprets the data during three 7 hour shifts per day.

SITE SELECTION AND TRANSDUCER DEPLOYMENT

The gently-sloping river bottom and small cobble at the historic right bank counting location has proven adequate for ensonification. A detailed bottom profile was obtained after initial transducer placement at the counting location by stretching a rope across the river and measuring water depth with a pole every meter. The transducer and automatic rotator were mounted on a pod made of aluminum pipe and deployed from the right-bank point bar. The pod was secured in place with sandbags and designed to permit raising and lowering the transducer by sliding it up or down along 2 riser pipes that extended above the water. Fine adjustments were made with remote control of the dual-axis rotator attached to the transducer. The transducer was deployed in water ranging from approximately 0.5 m to 1.0 m in depth, and aimed perpendicular to the current along the natural gravel substrate. An attempt was made to ensure the transducer was deployed at locations where minimum surface water velocities did not fall below 30–45 cm/s.

The system operator used an artificial acoustic target during deployment to ensure transducer aim was low enough to prevent salmon from passing undetected beneath the acoustic beam. The target, an airtight 250 ml weighted plastic bottle, was allowed to drift downstream along the river bottom and through the acoustic beam. Several drifts were made with the target in an attempt to pass it through as much of the counting range as possible. Proper transducer aim was verified with visual interpretation (echogram) on a computer screen as well as the oscilloscope. Later in the season, a 1.5 inch tungsten carbide sphere was used to verify how close to the bottom we could detect the target.

As in previous years, a fish lead was constructed shoreward from the transducer to prevent upstream salmon passage inshore of the transducer. The fish lead was constructed using 5 cm by 5 cm by 1.2-m high galvanized chain-link fencing and 2.5 m metal "T" stakes. The lead was constructed to include the nearfield of the sonar transducer. Whenever the transducer was relocated because of rising or falling water level, the lead was shortened or lengthened as appropriate, and the artificial target used to ensure proper re-aiming.

SONAR COUNT ADJUSTMENTS

At the end of each day, data collected by the DEP in 24 hourly text files was transferred to another computer for tracking and editing. To facilitate tracking, echoes from stationary objects were removed using a custom program created in *Java* computer language (Appendix B1). The data was manually edited to remove spurious tracks such as those from remaining bottom using *Polaris,* an echogram editor developed by Mr. Peter Withler through a cooperative agreement with Department of Fisheries and Oceans Canada (DFO), ADF&G, and HTI. Fish tracks were then manually counted using *Polaris*. Hourly estimates were exported to a *Microsoft Excel* spreadsheet where expansion or linear interpolation was used for periods of missing data.

STATIONARY BOTTOM REMOVAL

Echoes from stationary objects were removed before tracking by dividing data into range bins (0.2 meters), calculating the moving average (averaging window of 1,000 echoes) of the voltage in each range bin and then removing the echo if the voltage was within 1.7 standard deviations of the mean and at least 100 echoes were within that range bin. The echo was not removed if the percentage of missed echoes relative to observed echoes was greater than 80. The percentage of missed relative to observed echoes was calculated by summing differences between observed ping numbers minus 1 and then dividing by the total number of echoes in the range bin.

TRACKING

After the data was edited with the bottom removal program, the operator selected groups of echoes considered to be upstream fish based on visual interpretation of the *Polaris* echogram. These echoes grouped into fish tracks can be enumerated to produce an estimate of fish passage. Three times a day the crew saved an hour of tracked data to determine range distribution of the passing fish.

FINAL EDITING

The counts from each sample were entered into a *Microsoft Excel* spreadsheet where linear interpretation or expansion was used when data could not be collected due to relocating the transducer, system failure, or other unforeseen circumstances. When a portion of an hour (if 10 min or more) was missing, the hourly passage was estimated by dividing the count by the fraction of the hour sampled. In instances where complete hours were missed, linear interpolation was used. For example, when an entire 60-min sample was missing, counts were interpolated by averaging counts from 2 hours before and 2 hours after the missing period. If counts for 2 samples were missing, the counts from the 3 hours before and 3 hours after the missing samples were averaged, and so on. Sonar counts caused by fish other than salmon were assumed insignificant based upon historic visual “tower” observations and test fishing records collected at the site. After editing was complete the final estimate of hourly, daily, and cumulative fish passage was produced.

TEST FISHING AND SALMON SAMPLING

Region-wide standards have been set for the sample size needed to describe the age composition of a salmon population. These standards apply to the period or stratum in which the sample is collected. Sample size goals are based on a one-in-ten chance (precision) of not having the true age proportion (p_i) within the interval $p_i \pm 0.05$ for all i ages (accuracy).

As described in Bromaghin (1993), a sample size of 150 chum salmon is needed, assuming 2 major age classes with minor ages pooled, and no unreadable vertebrae. The preferred method of aging Yukon River fall chum salmon, when in close proximity to their natal streams, is from vertebrae collections (Clark 1986). Allowing for 20% unreadable vertebrae, the Sheenjek River sample size goal was to sample approximately 30 chum salmon per week up to a maximum of 180.

An adult salmon beach seine was periodically fished at different locations between the sonar site and approximately 10–18 km upstream to collect adult salmon for age and sex composition. The beach seine (3-inch stretch measure) was 30 m in length by 55 meshes deep (~3 m). The seine was dyed green, constructed of #18 twine, possessed 3- by 5-inch high-density, non-grommet oval poly floats spaced approximately 45 cm apart, had a 115–120 lb lead line and 1/2 in (1.3 cm) float line. Chum salmon were collected with the beach seine, enumerated by sex using external characteristics, and measured in millimeters from mideye to tail fork. Additionally, 3 vertebrae were taken from each fish for age determination.

CLIMATOLOGICAL AND HYDROLOGICAL OBSERVATIONS

A water level gauge was installed at the sonar site and monitored daily with readings made to the nearest centimeter. Surface water temperature was measured daily with a pocket thermometer. Minimum and maximum air temperatures, and wind velocity and direction were measured daily with a Weather Wizard III weather station. Other daily observations included recording occurrence of precipitation and estimating percent cloud cover. Climatological observations were recorded at approximately 1900 hours daily.

DIDSON™ Deployment and Operation

DIDSON™ was operated side-by-side with the split-beam system on the right bank of the Sheenjek River from August 18 through September 24, 2004. The DIDSON™ is a dual frequency, multi-beam sonar developed by the University of Washington, Applied Physics Laboratory. DIDSON™ produces images that are near video quality, allowing the operator to distinguish upstream fish from downstream fish and debris. This sonar allows transducer placement in areas that were not possible with other systems, such as areas with large rocks, submerged vegetation, or uneven bottom. The DIDSON™ was deployed using the same type of pod as the split-beam transducer. Attached to the transducer was an HTI model 662H dual-axis rotator with HTI model 660 remote controller to facilitate aiming. The electronic equipment was kept in the same tent as the split-beam equipment and powered with the same 1000 watt generator. The DIDSON™ was operated from 0300 to 0400 hours, 1400 to1500 hours, and 2100 to 2200 hours daily. Fish passage estimates were calculated using four 15-min samples from each hour of operation. Only hours with 60 minutes of data were used for comparison with the split-beam system. The system operator manually counted the fish from an echogram produced by the DIDSON™ system. DIDSON™ counts were then adjusted for range, to compare with the range ensonified by the split-beam system.

RESULTS

River and Sonar Counting Conditions

In 2004, location of transducer deployment approximated the same place on the point bar used in recent years. The river bottom at the counting location sloped gently from the convex bank (right-bank, point bar) at a rate of approximately 10 cm/m (bottom slope ≈ 10%) to the shelf-break that lay approximately two-thirds of the way across the channel on August 10 (Figure 5). River width measured 47 m and much of the nearshore zone along the concave, left cutbank, was cluttered with fallen trees and other woody vegetation.

The water level remained relatively low at the project site through 2004, with the lowest level recorded on September 25 (Figure 6; Appendix C1). With respect to the initial reading of the water gauge upon deployment on August 8, the water level fell 19.0 cm during the first 8 days then leveled off until August 23. The water level dropped continuously during the remainder of the project. Final measurement on September 25 was 59.0 cm below the initial level. Water temperature at the project site ranged from 3.0°C to 17.2°C based upon instantaneous surface measurements, and averaged 9.9°C (Appendix C1). The Porcupine and Sheenjek rivers were both beginning to freeze during the final days of operation.

Fluctuations in water level affected placement of the transducer with respect to shore, and in turn the proportion of the river ensonified. While no attempt was made to estimate fish passage beyond the counting range, occasional expansions or interpolations of sonar counts were made to estimate fish passage for periods when data was missing because of system failures or moving the transducer.

ABUNDANCE ESTIMATION

The 2004 sonar-estimated escapement was 37,878 chum salmon for the 49-day period August 8 through September 25 (Table 3). As in the past, only estimates from the HTI sonar on the right bank were used to produce the estimated escapement of fall chum salmon. Fish were counted from the data files during each shift and adjustments to the equipment or data was made if necessary. Daily passage estimates were relayed to the fishery managers in Fairbanks every morning via satellite telephone.

TEMPORAL AND SPATIAL DISTRIBUTION

Chum salmon were present in the river when sonar counting was initiated at 1300 hours on August 8, as evidenced by the 281 fish estimated passing that day. Three distinct pulses of chum salmon passed the sonar in 2004 (Figure 7), the largest passage estimate of 3,124 fish occurring on September 21. The middle portion of the run was observed from September 5 through September 20, the median day of passage occurred on September 13. The average passage rate during this period approximated 1,365 fish per day. An estimated 891 chum salmon passed the project site on September 25, the final day of sonar operation. Factors affecting termination of sonar counting in 2004 included logistics associated with closing down camp, and impending winter weather.

The diel pattern of migration of Sheenjek River chum salmon typically observed on the right bank in most years (Dunbar 2004) was again manifested in 2004 (Figure 8). Upstream migration was heaviest in periods of darkness or suppressed light, with fish moving in greater numbers close to shore. On average, the periods of greatest upstream migration occurred at 0900 hours and 2100 hours. The period of least movement in 2004 was 1300 hours.

Most migrating chum salmon were shore-oriented, passing through the nearshore portion of the acoustic beam. Approximately 85% of the fish counted were estimated passing through the first 11 m of the counting range. The first few meters had fewer fish due to the placement of the fish lead in relation to the transducer. The offshore half of the counting range was 15% of the total, only 0.24% was observed in the outer-most 3 meters (Figure 9).

AGE AND SEX COMPOSITION

Although an attempt was made to sample portions of annual escapement for age and sex composition in 2004, only 107 chum salmon (66 males; 41 females) were obtained due to distribution and availability of salmon for sampling (Table 4). Twenty-six seine hauls were made during the period September 5 through September 23 along gravel bars between river kilometers (rkm) 10 and 18. Although the overall sample goal had not been met, sampling was terminated on September 23 because of camp breakdown activities, and the late portion of the run was sufficiently sampled. Three of the 107 vertebrae collected were unreadable. From the remaining 104 samples it was determined that age-4 predominated (61.5%), the proportion of age-5 fish observed was 25.0%, age-3 fish was 11.5%, and age-6 fish was 1.9% (Appendix D1).

SPLIT-BEAM AND DIDSON™ COMPARISON

Comparison of the DIDSON™ and split-beam sonar estimates was conducted during periods of low and moderately high passage during the period of August 18–September 24. During this 38-day period, 105 complete, paired 1-hour samples were collected. As with the split-beam, most fish were observed relatively close to shore, with passage greatest during hours of darkness or suppressed light. Approximately 83% of the fish counted were estimated passing through the first 11 m of the counting range (Figure 10). The first few meters had fewer fish because of placement of the DIDSON™, about 1 meter behind the split-beam transducer, and the location of the fish lead in relation to the DIDSON™. The offshore half of the counting range was 17% of the total, while only 1% was observed in the outer-most 3 meters.

DISCUSSION

ESCAPEMENT ESTIMATE

The 2004 sonar-estimated escapement of chum salmon in the Sheenjek River is considered conservative because fish that passed the site before sonar sampling were not included, nor were fish that passed beyond the range of the acoustic beam, including along the left bank. Because of information collected with the DIDSON™ in 2003 suggesting that 33% of the fish migrate up the left side of the river (Dunbar 2006), it is anticipated that ADF&G will operate a DIDSON™ on both banks in 2005.

Although sonar has been used to monitor chum salmon escapements in the Sheenjek River since 1981, project operational dates have only been consistent since 1991. Barton (1995) used run timing data collected from the nearby Chandalar River to expand Sheenjek River run size estimates for the years 1986–1988, and 1990 to a comparable time period. The 1989 estimate was expanded from aerial survey observations made before sonar operations in that year (Table 2). Barton (2002) used historic run timing data from 1986 to 1999 to expand the estimated escapement for 2000, when sonar operations terminated early. From average run timing data for 1986–2003, approximately 85% of the Sheenjek River fall chum salmon run (through the end of September) materializes subsequent to August 25, with the middle portion of the run passing from August 31 through September 17. The historical median day of passage is September 9. Although fish were present in the river early, most fish arrived later; the median passage day in 2004 was 4 days later than the historical average.

The escapement estimate in 2004 approximated 37,878 chum salmon for the 49-day period of August 8–September 25. This escapement estimate was the seventh lowest recorded at Sheenjek River, and was not enough to meet the low end of the revised BEG of 50,000 to 104,000 chum salmon (Figure 11). This low run was somewhat expected because the major parent year escapement levels were 14,229 in 1998 (returning age-5 fish) and 30,084 in 1999 (returning age-4 fish).

Low numbers of returning fall chum salmon were also reported in the Fishing Branch River, where 20,274 chum salmon passed the DFO weir during the 50-day period of August 22 through October 10 (JTC 2005). The Fishing Branch River escapement was above the interim escapement goal of 13,000, but well below the normal escapement goal range of 50,000 to 120,000 fish.

The 2004 season was characterized by above average even-year fall chum salmon runs to most Yukon drainage river systems. Since 1997, the upper drainage fall chum salmon production appears to be driven by the continued relative strength of the Chandalar River and mainstem Yukon River components. The Porcupine drainage being an exception, with low fall chum salmon returns to both the Sheenjek and Fishing Branch rivers. Most fall chum salmon BEGs were achieved within the Yukon River drainage in 2004, commercial fishing was limited only by market conditions and buyer interest, and subsistence restrictions were not necessary except in the Canadian portion of the Porcupine drainage.

SPLIT-BEAM AND DIDSON™ COMPARISON

Diel patterns were similar with both systems. More fish were counted at night and periods of low light than were counted during daylight hours. Spatial distribution was about the same with both systems. The DIDSON™ was well suited for conditions experienced on the Sheenjek River, and proved to be easier to operate and more accurate than the split-beam sonar. Further comparison and analysis of the 2 systems will be completed and published in a future report.

ACKNOWLEDGMENTS

The author wishes to acknowledge the sonar field camp personnel, ADF&G technicians Susan Klock, Elizabeth Smith and Yukon River Drainage Fisheries Association technician Gary Weber for their dedication to the project and collecting most of the data used in this report. Thanks to Carl Pfisterer and Bruce McIntosh for logistical support. Finally, I thank Eric Volk, Toshihide Hamazaki, and Carl Pfisterer for their review and editorial comments on this manuscript. This investigation was partially funded by U.S./Canada Salmon Research Cooperative under NOAA Cooperative Agreement Award Number NA04NMF4380264.

REFERENCES CITED

Barton, L. H. 1984. A catalog of Yukon River salmon spawning escapement surveys. Alaska Department of Fish and Game, Division of Commercial Fisheries, Technical Data Report No. 121, Juneau.

Barton, L. H. 1985. Enumeration of fall chum salmon by side-scanning sonar in the Sheenjek River in 1984. Alaska Department of Fish and Game, Division of Commercial Fisheries, Yukon Salmon Escapement Report No. 25, Fairbanks.

Barton, L. H. 1995. Sonar enumeration of fall chum salmon on the Sheenjek River, 1988-1992. Alaska Department of Fish and Game, Commercial Fisheries Management and Development Division, Technical Fishery Report 95-06, Juneau.

Barton, L. H. 2002. Sonar estimation of fall chum salmon abundance in the Sheenjek River, 2000. Alaska Department of Fish and Game, Division of Commercial Fisheries, Regional Information Report 3A02-26, Anchorage.

Bromaghin, J. F. 1993. Sample size determination for interval estimation of multinomial probabilities. The American Statistician 47 (3):203-206.

Buklis, L. S. 1993. Documentation of Arctic-Yukon-Kuskokwim region salmon escapement goals in effect as of the 1992 fishing season. Alaska Department of Fish and Game, Division of Commercial Fisheries, Regional Information Report 3A93-03, Anchorage.

Buklis, L. S. and Barton, L. H. 1984. Yukon River fall chum salmon biology and stock status. Alaska Department of Fish and Game, Division of Commercial Fisheries, Informational Leaflet No. 239, Juneau.

Clark, R. A. 1986. Sources of variability in three ageing structures for Yukon River fall chum salmon (*Oncorhynchus keta* Walbaum) escapement samples. Alaska Department of Fish and Game, Division of Sport Fish, (Region III unpublished report), Fairbanks.

Dunbar, R. D. 2004. Sonar estimation of fall chum salmon abundance in the Sheenjek River, 2002. Alaska Department of Fish and Game, Division of Commercial Fisheries, Regional Information Report 3A04-10, Anchorage.

Dunbar, R. 2006. Sonar estimation of fall chum salmon abundance in the Sheenjek River, 2003. Alaska Department of Fish and Game, Fishery Data Series No. 06-65, Anchorage. http://www.sf.adfg.state.ak.us/FedAidPDFs/fds06-65.pdf

Eggers, D. M. 2001. Biological escapement goals for Yukon River fall chum salmon. Alaska Department of Fish and Game, Commercial Fisheries Division, Regional Information Report 3A01-10, Anchorage.

HTI (Hydroacoustic Technology Incorporated). 2000. Model 241/243/244/Split-Beam Digital Echo Sounder System Operators Manual Version 1.8. Hydroacoustic Technology Incorporated, Seattle.

JTC (Joint Technical Committee of the Yukon River US/Canada Panel). 2005. Yukon River Salmon 2004 season summary and 2005 season outlook. Fisheries and Oceans Canada, Stock Assessment and Fisheries Management Section, Yukon and Transboundary Area. Whitehorse, Yukon Territory.

Maxwell, S. L., and N. E. Gove. 2004. The feasibility of estimating migrating salmon passage in turbid rivers using a Dual Frequency Identification Sonar (DIDSON), 2002. Alaska Department of Fish and Game, Division of Commercial Fisheries, Regional Information Report 2A04-05, Anchorage.

Seeb, L. W., P. A. Crane, and R. B. Gates. 1995. Progress report of genetic studies of Pacific Rim chum salmon and preliminary analysis of the 1993 and 1994 South Unimak June fisheries. Alaska Department of Fish and Game, Commercial Fisheries Management and Development Division, Regional Information Report 5J95-07, Juneau.

Wilmot, R. L., R. J. Everett, W. J. Spearman, and R. Baccus. 1992. Genetic stock identification of Yukon River chum and Chinook salmon 1987 to 1990. Progress Report, U.S. Fish and Wildlife Service, Anchorage.

TABLES AND FIGURES

Table 1.–Alaskan and Canadian total utilization of Yukon River fall chum salmon, 1970–2004.

Year	Canada [a]	Alaska [b,c]	Total
1970	3,711	265,096	268,807
1971	16,911	246,756	263,667
1972	7,532	188,178	195,710
1973	10,135	285,760	295,895
1974	11,646	383,552	395,198
1975	20,600	361,600	382,200
1976	5,200	228,717	233,917
1977	12,479	340,757	353,236
1978	9,566	331,250	340,816
1979	22,084	593,293	615,377
1980	22,218	466,087	488,305
1981	22,281	654,976	677,257
1982	16,091	357,084	373,175
1983	29,490	495,526	525,016
1984	29,267	383,055	412,322
1985	41,265	474,216	515,481
1986	14,543	303,485	318,028
1987	44,480	361,663 [d]	406,143
1988	33,565	319,677	353,242
1989	23,020	518,157	541,177
1990	33,622	316,478	350,100
1991	35,418	403,678	439,096
1992	20,815	128,031 [e]	148,846
1993	14,090	76,925 [d]	91,015
1994	38,008	131,217	169,225
1995	45,600	415,547	461,147
1996	24,354	236,569	260,923
1997	15,580	154,479 [e]	170,059
1998	7,901	62,869 [d]	70,770
1999	19,506	110,369	129,875
2000	9,236	19,307 [d]	28,543
2001	9,512	35,154 [d]	44,666
2002	8,018	19,393 [d]	27,411
2003	11,355	68,174	79,529
2004	9,750	4,110	13,860
Average			
1970–03	20,268	286,385	306,652
1979–83	22,433	513,393	535,826
1994–03	18,907	125,308	144,215
1999–03	11,525	50,479	62,005

Source: JTC 2005.

[a] Catch in number of salmon. Includes commercial, Aboriginal, domestic and sport catches combined.

[b] Catch in number of salmon. Includes estimated number of salmon harvested for commercial production of salmon roe.

[c] Commercial, subsistence, personal-use and ADF&G test fish catches combined.

[d] Commercial fishery did not operate in Alaskan portion of drainage.

[e] Commercial fishery operated only in District 6 (Tanana River).

Table 2.–Operational dates, and escapement estimates of fall chum salmon in the Sheenjek River, 1981–2004.

Year	Starting Date	Ending Date	Project Duration	Sonar Estimate	Expanded Estimate
1981	31-Aug	24-Sep	25	74,560	
1982	31-Aug	22-Sep	23	31,421	
1983	29-Aug	24-Sep	27	49,392	
1984	30-Aug	25-Sep	27	27,130	
1985	02-Sep	29-Sep	28	152,768	
1986	17-Aug	24-Sep	39	83,197[a]	84,207
1987	25-Aug	24-Sep	31	140,086	153,267
1988	21-Aug	27-Sep	38	40,866	45,206
1989	24-Aug	25-Sep	33	79,116	99,116
1990	22-Aug	28-Sep	38	62,200	77,750
1991	09-Aug	24-Sep	47	86,496	
1992	09-Aug	20-Sep	43	78,808	
1993	08-Aug	28-Sep	52	42,922	
1994	07-Aug	28-Sep	53	150,565	
1995	10-Aug	25-Sep	47	241,855	
1996	30-Jul	24-Sep	57	246,889	
1997	09-Aug	23-Sep	46	80,423	
1998	17-Aug	30-Sep	45	33,058	
1999	10-Aug	23-Sep	45	14,229	
2000	08-Aug	12-Sep	36	18,652[b]	30,084
2001	11-Aug	23-Sep	44	53,932	
2002	09-Aug	24-Sep	47	31,642	
2003	09-Aug	26-Sep	49	38,321[c]	44,047
2004	08-Aug	25-Sep	49	37,878	
1981-85	30-Aug	24-Sep	26	67,054	
1986-90	21-Aug	25-Sep	36	81,093	91,909
1991-00	08-Aug	23-Sep	47	99,390	100,533
1999-03	09-Aug	21-Sep	44	31,355	34,787

[a] Sonar–estimated escapement in these years was subsequently expanded to include fish passing prior to sonar operations (Barton 1995). Expansions for 1986–1988 and 1990 were based upon run timing data collected in the nearby Chandalar River. The 1989 estimate was expanded based upon aerial survey observations made in the Sheenjek River prior to sonar operations in that year.

[b] Sonar-estimated escapement was expanded to include fish passing after sonar operations terminated (Barton 2002). Expansions for 2000 were based upon average run time data from the Sheenjek River 1986–1999.

[c] Sonar-estimated escapement was expanded to include fish passing after sonar operations terminated. Expansions for 2003 were based upon run time data from the Rampart tag recovery fish wheel.

Table 3.–Sonar-estimated passage of fall chum salmon in the Sheenjek River, 2004.

	Number of Salmon		Proportion	
Date	**Daily**	**Cumulative**	**Daily**	**Cumulative**
08-Aug	281	281	0.01	0.01
09-Aug	393	674	0.01	0.02
10-Aug	300	974	0.01	0.03
11-Aug	251	1,225	0.01	0.03
12-Aug	243	1,468	0.01	0.04
13-Aug	182	1,650	0.00	0.04
14-Aug	152	1,801	0.00	0.05
15-Aug	146	1,947	0.00	0.05
16-Aug	167	2,114	0.00	0.06
17-Aug	191	2,305	0.01	0.06
18-Aug	178	2,483	0.00	0.07
19-Aug	268	2,751	0.01	0.07
20-Aug	221	2,972	0.01	0.08
21-Aug	200	3,173	0.01	0.08
22-Aug	157	3,329	0.00	0.09
23-Aug	220	3,549	0.01	0.09
24-Aug	183	3,732	0.00	0.10
25-Aug	120	3,852	0.00	0.10
26-Aug	189	4,041	0.00	0.11
27-Aug	174	4,215	0.00	0.11
28-Aug	334	4,550	0.01	0.12
29-Aug	375	4,925	0.01	0.13
30-Aug	284	5,209	0.01	0.14
31-Aug	529	5,738	0.01	0.15
01-Sep	735	6,473	0.02	0.17
02-Sep	779	7,252	0.02	0.19
03-Sep	523	7,775	0.01	0.21
04-Sep	721	8,496	0.02	0.22
05-Sep	1,538	10,034	0.04	0.26 [a]
06-Sep	641	10,676	0.02	0.28
07-Sep	829	11,505	0.02	0.30
08-Sep	843	12,348	0.02	0.33
09-Sep	1,238	13,586	0.03	0.36
10-Sep	920	14,506	0.02	0.38
11-Sep	1,954	16,460	0.05	0.43
12-Sep	1,470	17,930	0.04	0.47
13-Sep	1,775	19,705	0.05	0.52 [b]
14-Sep	1,830	21,535	0.05	0.57
15-Sep	1,226	22,761	0.03	0.60
16-Sep	1,389	24,150	0.04	0.64
17-Sep	908	25,058	0.02	0.66
18-Sep	1,283	26,341	0.03	0.70
19-Sep	1,613	27,953	0.04	0.74
20-Sep	2,377	30,330	0.06	0.80
21-Sep	3,124	33,454	0.08	0.88
22-Sep	972	34,427	0.03	0.91
23-Sep	1,834	36,261	0.05	0.96
24-Sep	725	36,986	0.02	0.98
25-Sep	891	37,877	0.02	1.00
Total	37,878		0.99	

[a] Single boxed area identifies central half of the run.

[b] Bold box identifies median day of passage.

Table 4.–Sheenjek River test fishing (beach seine) results, 2004.

Date	Number of Sets	Location (rkm)[a]	Chum Salmon Captured Male		Female		Total	Arctic Grayling
5-Sep	3	10, 11 & 13	3		0		3	3
10-Sep	4	18	5		1		6	3
11-Sep	3	18	14		5		19	1
13-Sep	3	18	8		3		11	6
19-Sep	3	18	2		1		3	4
20-Sep	6	18	17		19		36	2
23-Sep	4	18	17		12		29	4
Total	26		66	(62%)	41	(38%)	107	23

[a] Locations are river kilometer (rkm).

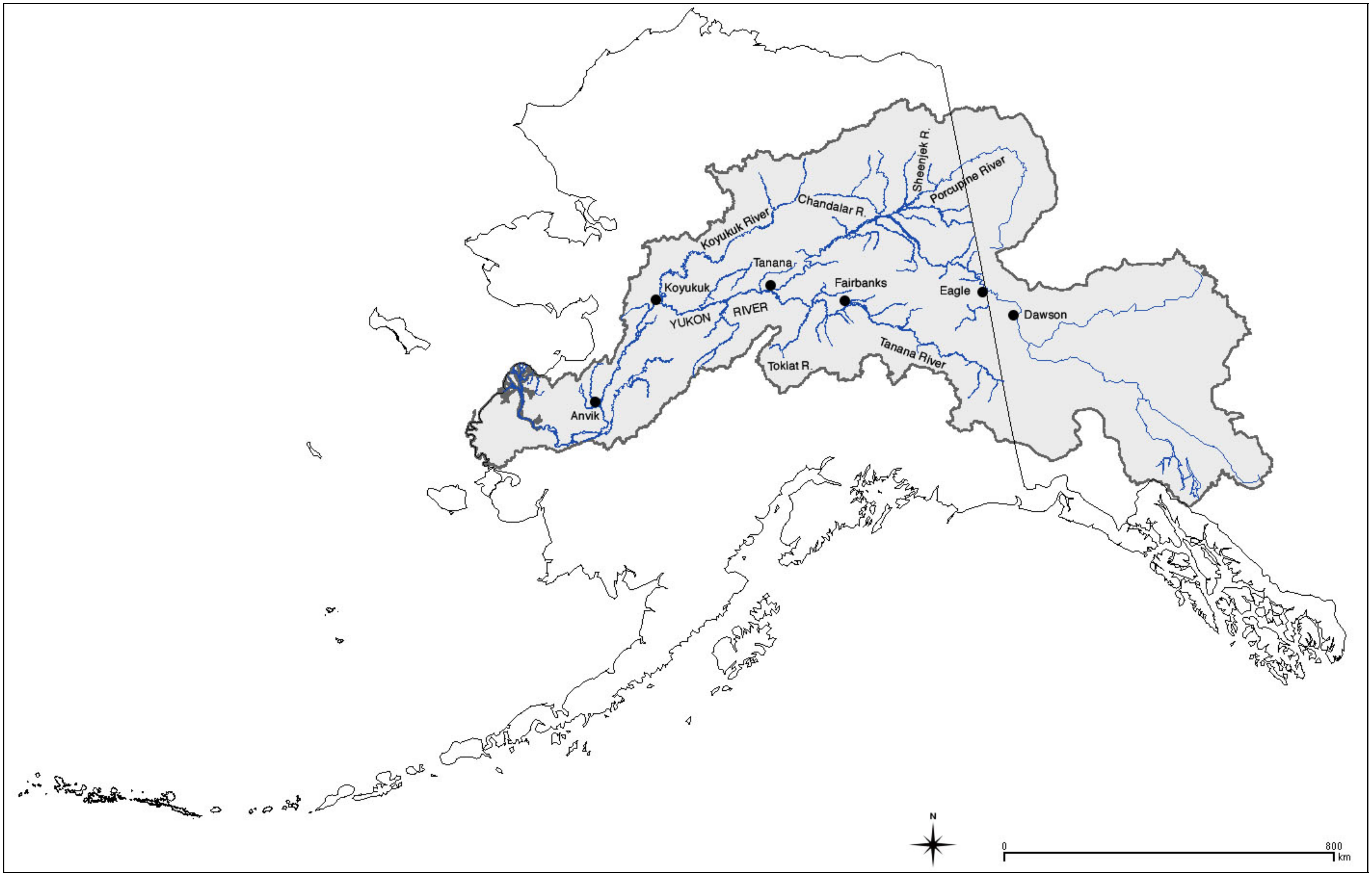

Figure 1.–Yukon River drainage showing selected locations.

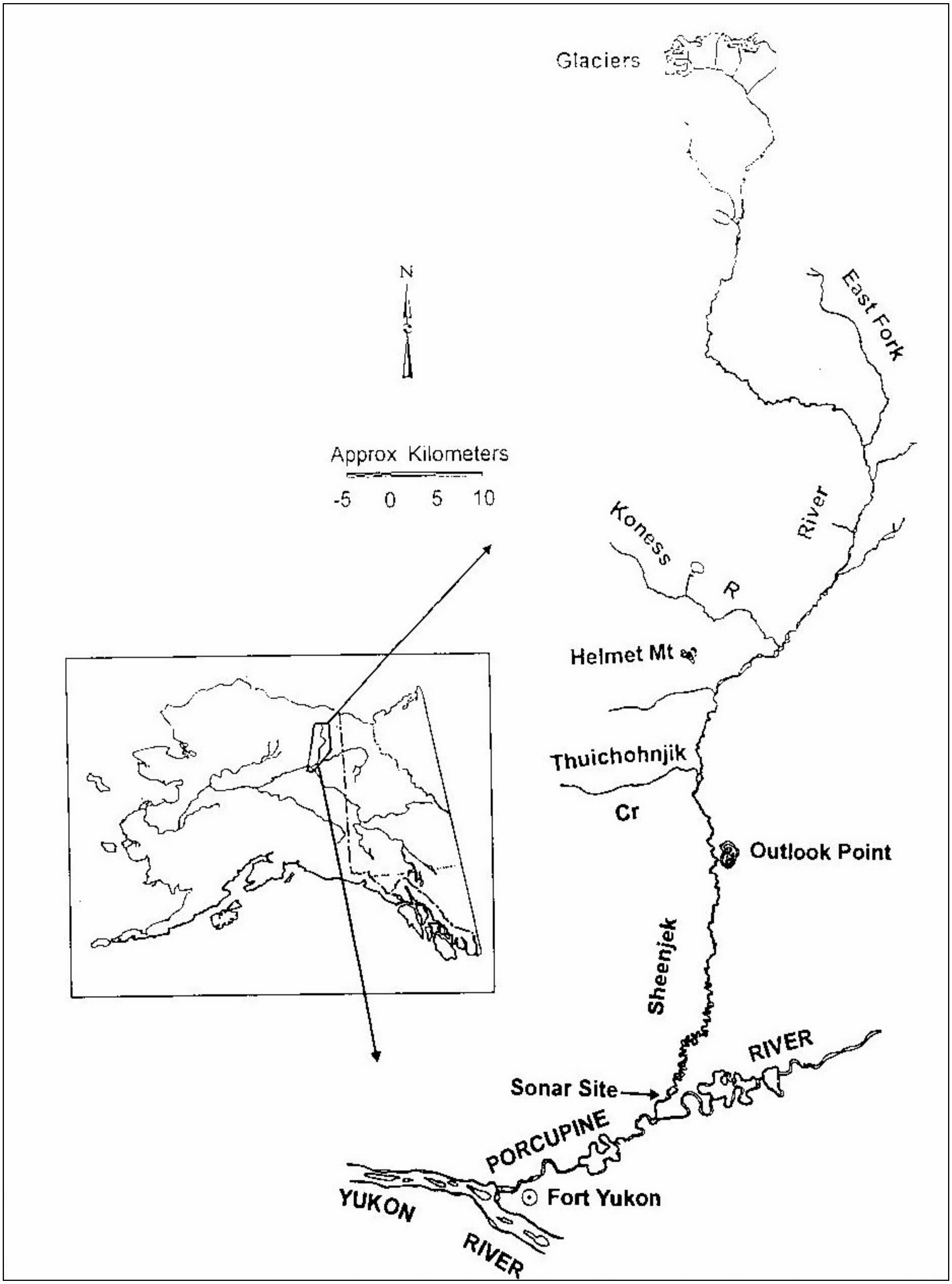

Figure 2.–Sheenjek River drainage.

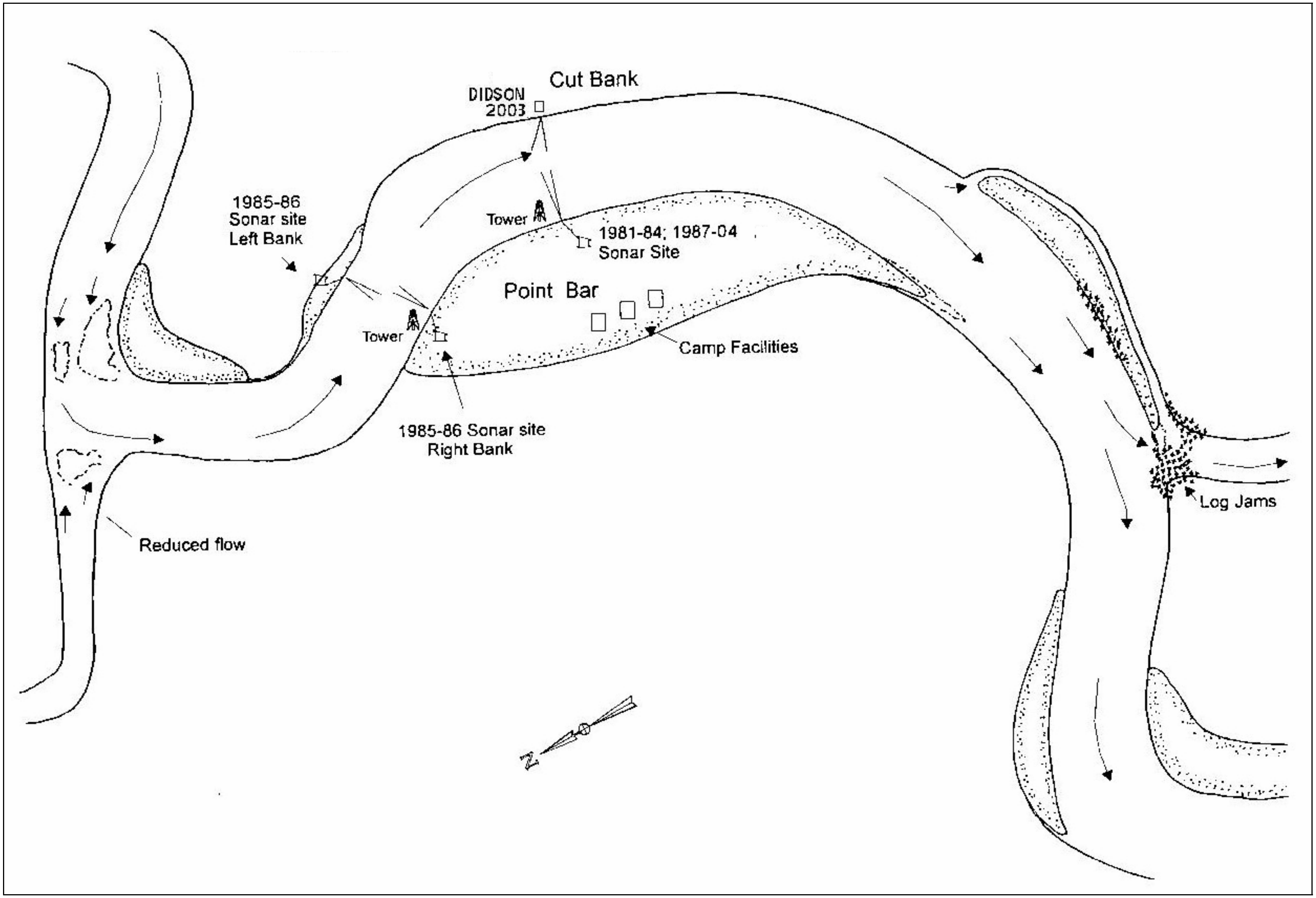

Figure 3.–Sheenjek River sonar project site.

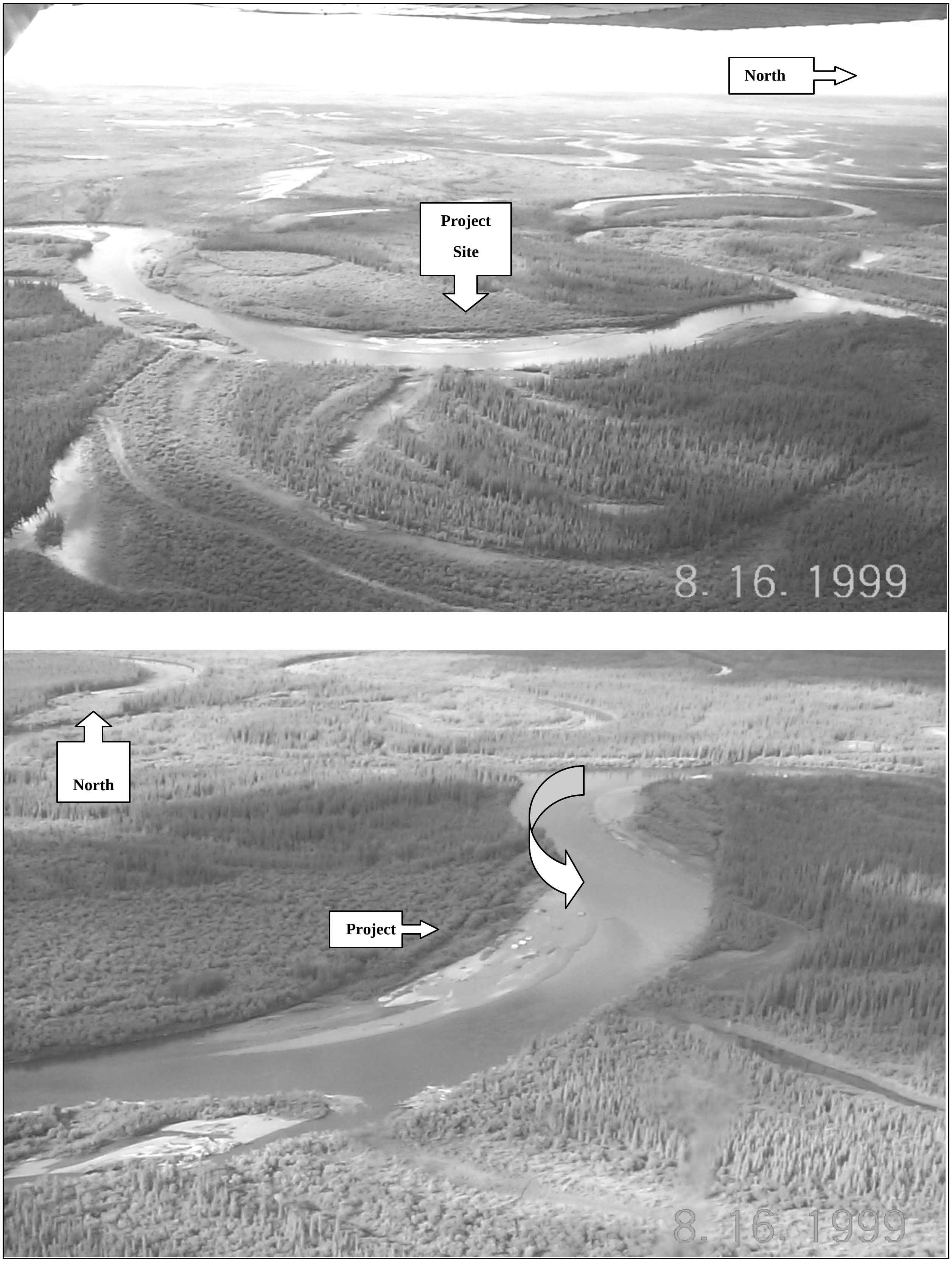

Figure 4.–Aerial photographs of the Sheenjek River sonar project site taken August 16, 1999.

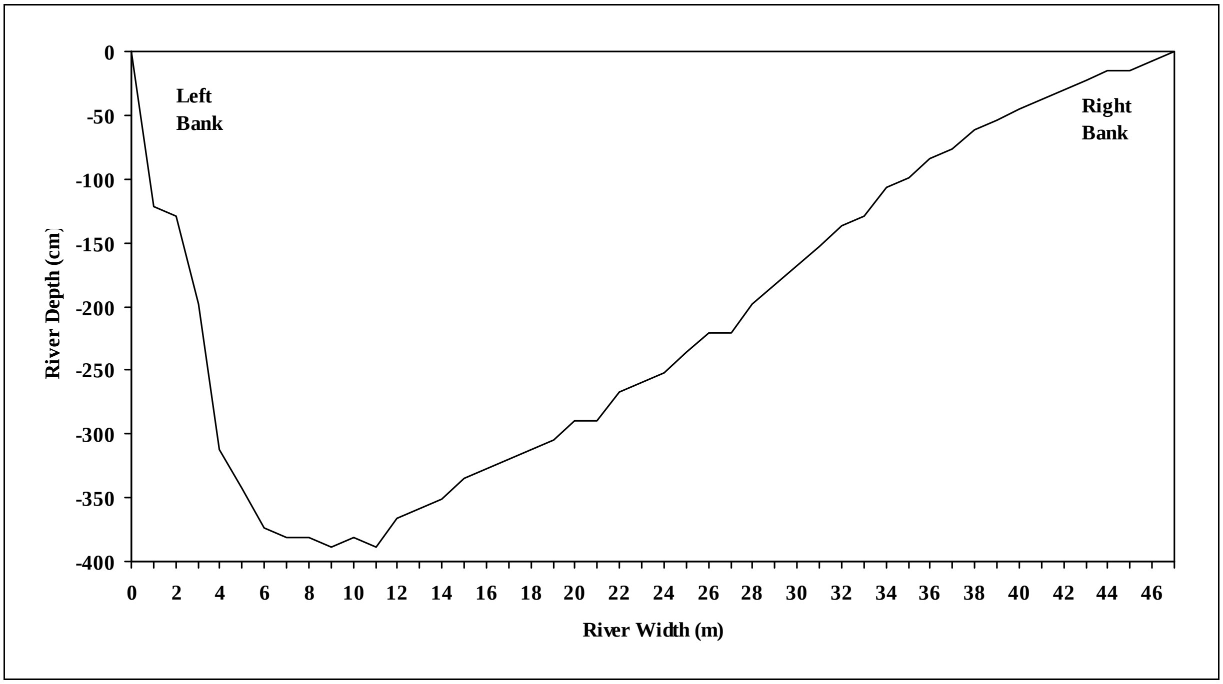

Figure 5.–Depth profile (downstream view) made August 10, 2004, at the Sheenjek River sonar project site.

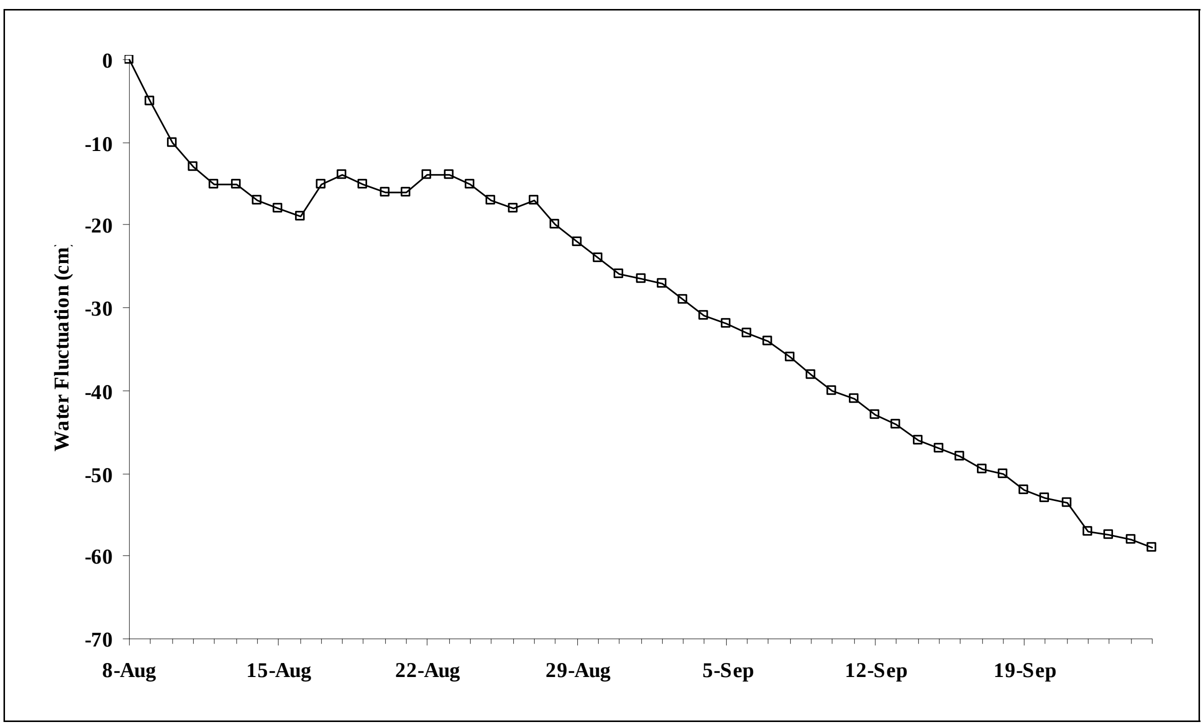

Figure 6.–Changes in daily water elevation relative to August 8 measured at the Sheenjek River sonar project site, 2004.

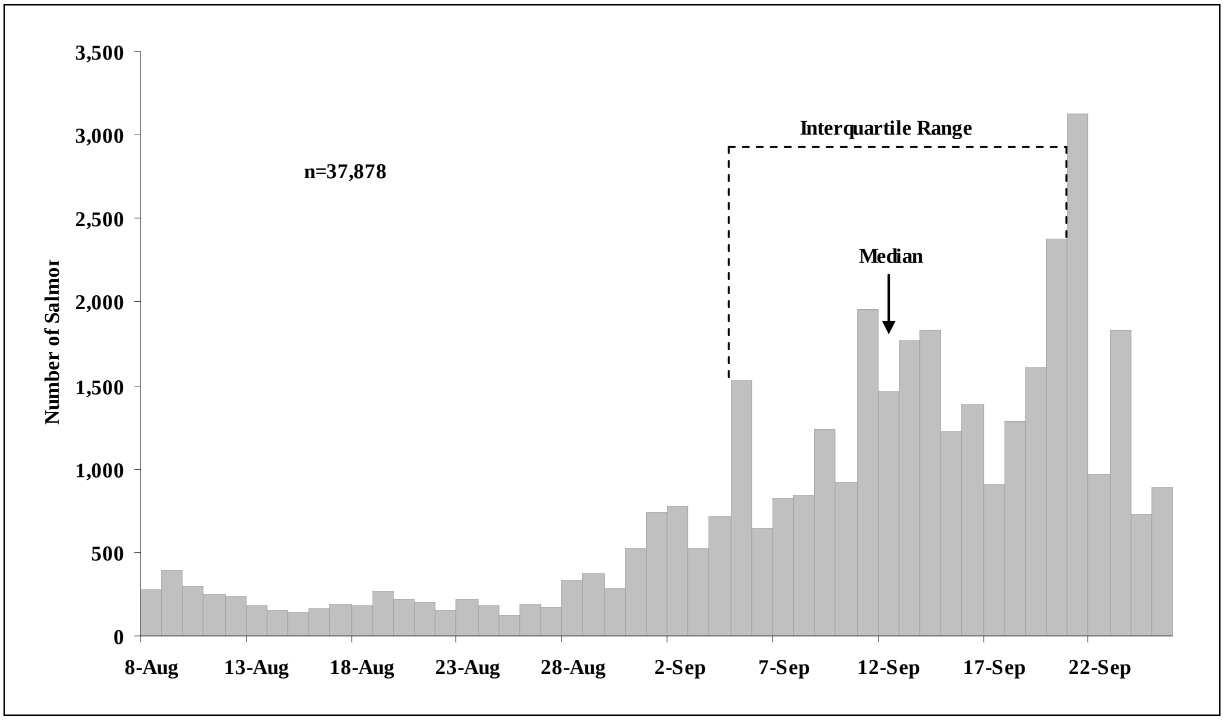

Figure 7.–Adjusted fall chum salmon sonar counts by date, Sheenjek River, 2004.

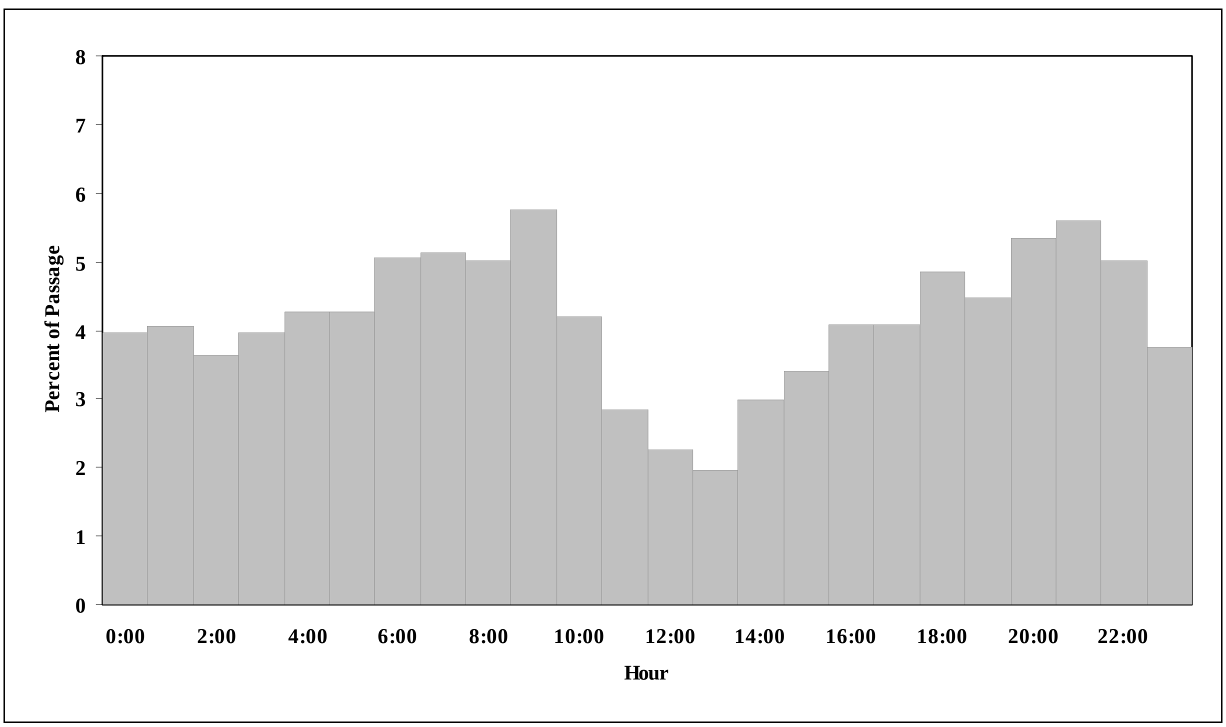

Figure 8.–Diel migration pattern of fall chum salmon observed on the right bank of the Sheenjek River, August 9–September 25, 2004.

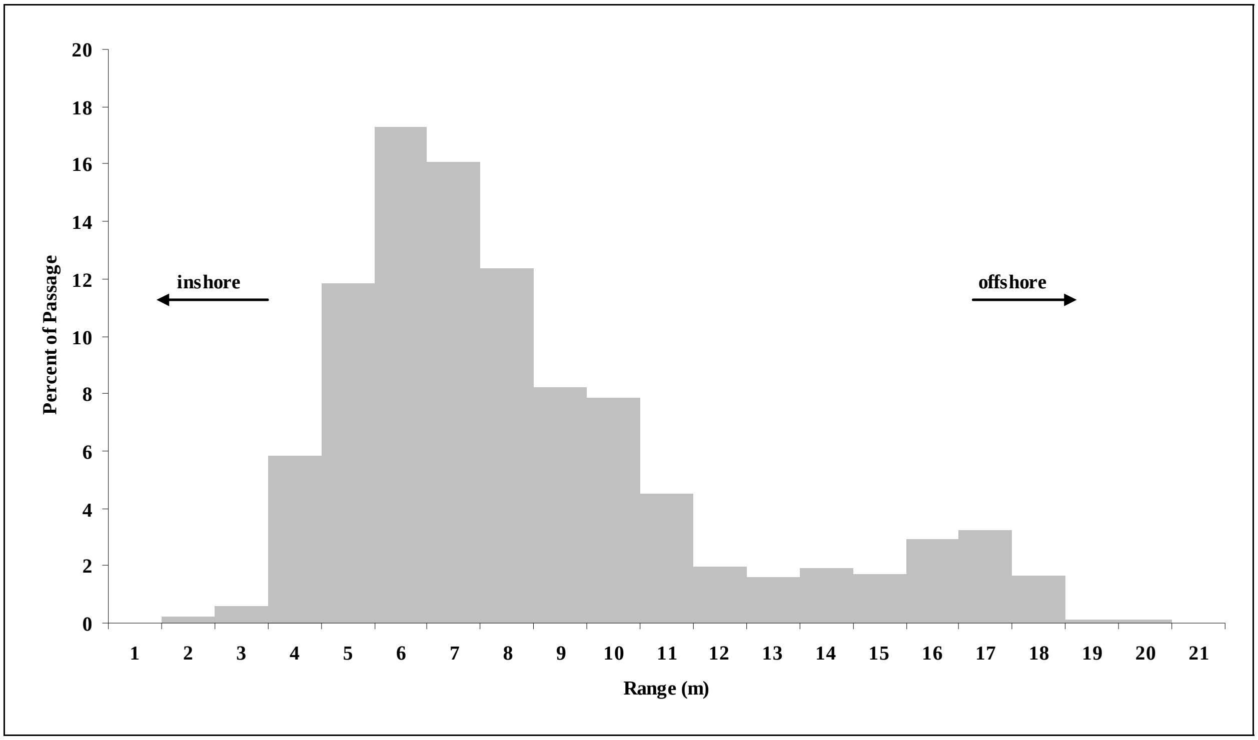

Figure 9.–Split-beam, right bank upstream fall chum salmon distribution in the Sheenjek River, 2004.

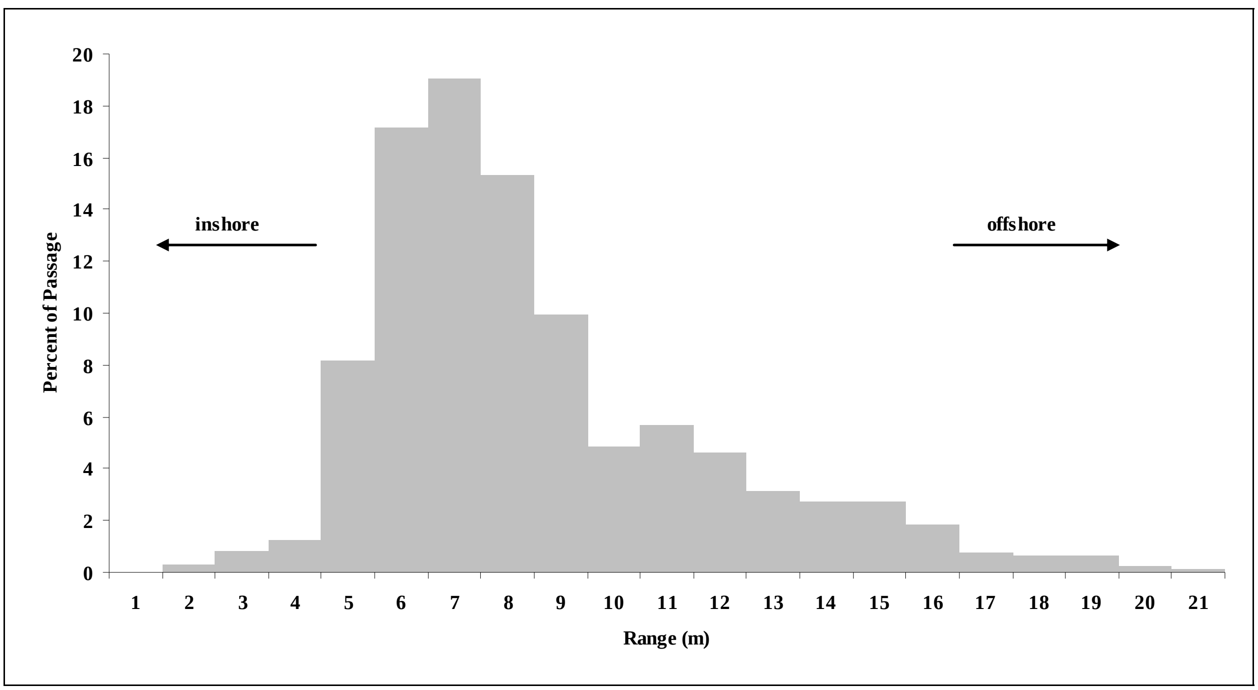

Figure 10.–DIDSON™, right bank upstream fall chum salmon distribution in the Sheenjek River, 2004.

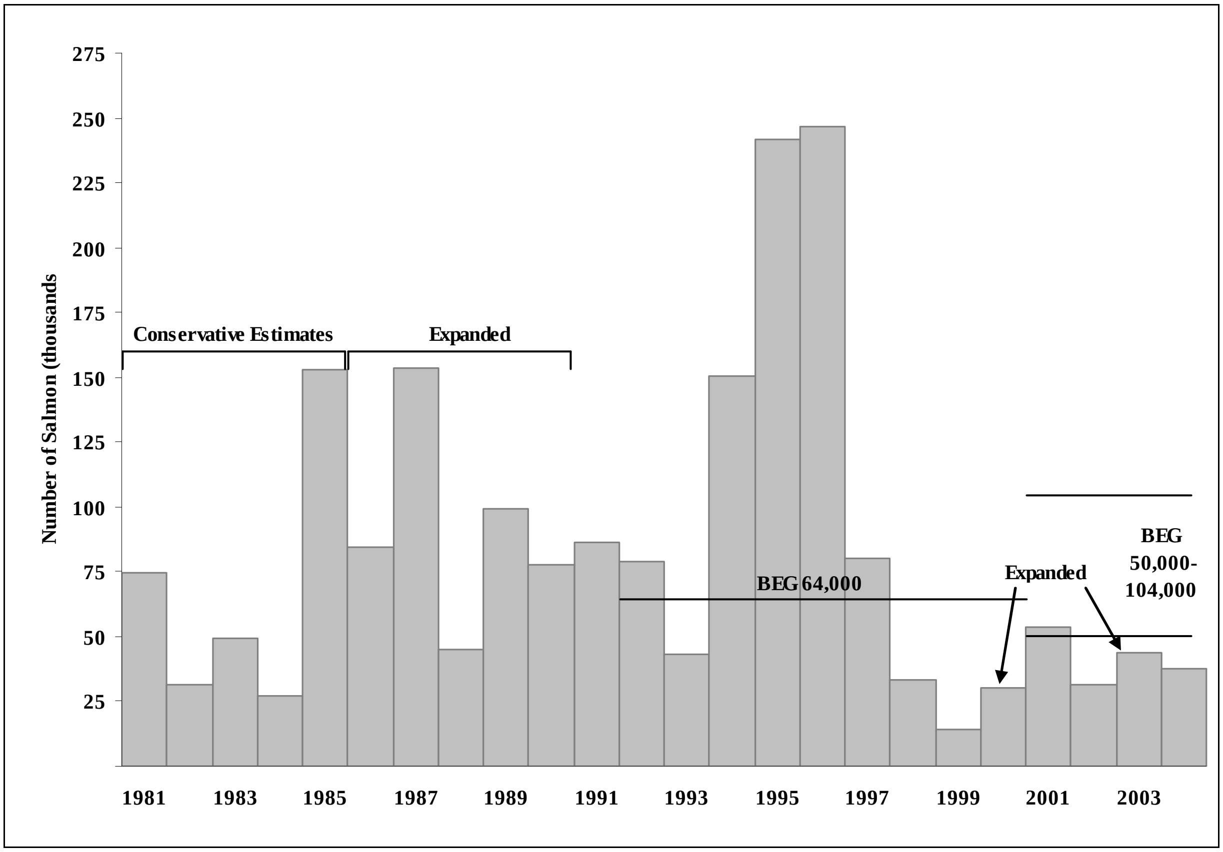

Figure 11.-Sonar-estimated escapement and BEG (horizontal lines) of fall chum salmon in the Sheenjek River, 1981–2004.

APPENDIX A. ECHO SOUNDER SPECIFICATIONS

Appendix A1.–Technical specifications for the Model 241 Portable Split-Beam Digital Echo Sounder.

Size:	10 inches wide x 4.3 high x 17 long, without PC or transducer (254 mm wide x 109 high x 432 long).
Weight:	20 lb. (9 kg) without PC or transducer.
Power Supply:	Nominal 12 VDC standard (120 VAC and 240 VAC optional).
Operating Temperature:	5-50°C (41-122°F).
Power Consumption:	30 watts (120 - 200 kHz), without laptop PC.
Frequency:	200 kHz standard (120 kHz and 420 kHz optional).
Transmit Power:	100 watts standard for 120-200 kHz. 50 watts standard for 420 kHz.
Dynamic Range:	140 dB
Transmitter:	Output power is adjustable in four steps over a 20 dBw range (+2, +8, +14, and 20 dBw).
Pulse Length:	Selectable from 0.1 msec to 1.0 msec in 0.1 msec steps.
Bandwidth:	Receiver bandwidth is automatically adjusted to optimize performance for the selected pulse length.
Receiver Gain:	Overall receiver gain is adjustable in five steps over a 40 dB range (-16, -8, 0, +8, +16 dB).
TVG Functions:	Simultaneous 20 and 40 log(R)+2αr TVG. Spreading loss and alpha are programmable to nearest 0.1 dB. Total TVG range is 80 dB. TVG start is selectable in 1m increments. The minimum TVG start is 1.0 m to maximum of 200 m.
Receiver Blanking:	Start and stop range blanking is selectable in 1m steps.
Undetected Output:	12 kHz, for each formed beam
Detected Output:	10 volts peak
System Synchronization:	Internal or external trigger
Ping Rate:	0.5-40.0 pings/sec
Phase Calculation:	Quadrature demodulation
Angular Resolution:	+/- <0.1° (6° beam width, 200 kHz)
Tape recording:	With Split-Beam Data Tape Interface and optional Digital Audio Tape (DAT) recorder, directly records the digitized split-beam data, permitting complete reconstruction of the raw data output.
Calibrator:	Local receiver calibration check using internal calibration source. Pulse and CW calibration functions provided in step settings.
Positioning:	GPS positioning information (NMEA 0183 format) via serial port of computer

Source: Model 241 operator's manual.

APPENDIX B. JAVA PROGRAM CODE

Appendix B1.–Java program code used to remove stationary object (bottom) echoes from HTI .raw echo files.

```
//
//  JBottomRemove.java
//          For information on setting Java configuration information, including setting
//          Java properties, refer to the documentation at
//                      http://developer.apple.com/techpubs/java/java.html
//

import java.awt.*;
import java.awt.event.*;
import java.util.*;
//import com.apple.mrj.*;
import javax.swing.*;
import javax.swing.filechooser.*;
import javax.swing.border.*;
import javax.swing.event.*;

import java.io.File;
import java.util.Hashtable;

public class JBottomRemove extends JFrame
                implements  ActionListener
                 //         MRJAboutHandler,
                  //        MRJQuitHandler
{

  //==========================
  //
  //  Declarations
  //
  //==========================
  static final String message = "   JBottomRemove";
  private Font font = new Font("serif", Font.ITALIC+Font.BOLD, 24);

  //protected AboutBox aboutBox;
```

-continued-

```
JButton filterButton = new JButton("Filter");

// Declarations for menus
static final JMenuBar mainMenuBar = new JMenuBar();

static final JMenu fileMenu = new JMenu("File");
protected JMenuItem miNew;
protected JMenuItem miOpen;
protected JMenuItem miClose;
protected JMenuItem miSave;
protected JMenuItem miSaveAs;

static final JMenu editMenu = new JMenu("Edit");
protected JMenuItem miUndo;
protected JMenuItem miCut;
protected JMenuItem miCopy;
protected JMenuItem miPaste;
protected JMenuItem miClear;
protected JMenuItem miSelectAll;
protected JMenuItem miParameters;

// Declarations for filtering parameters
protected double maxPingGap;
protected double windowWidth;
protected double critVal;
protected int averageWindow;
protected double threshold;
protected boolean softEdge;
protected double pingFraction;

protected File[] file;

//========================
//
//  End of declarations and
```

-continued-

```
//  start of methods
//
//============================

//------------------------------------------
//
//  Create the "File" menu
//
//------------------------------------------
public void addFileMenuItems() {
    miNew = new JMenuItem ("New");
    miNew.setAccelerator(KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_N,
java.awt.Event.META_MASK));
    fileMenu.add(miNew).setEnabled(false);
    miNew.addActionListener(this);

    miOpen = new JMenuItem ("Open...");
    miOpen.setAccelerator(KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_O,
java.awt.Event.META_MASK));
    fileMenu.add(miOpen).setEnabled(true);
    miOpen.addActionListener(this);

    miClose = new JMenuItem ("Close");
    miClose.setAccelerator(KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_W,
java.awt.Event.META_MASK));
    fileMenu.add(miClose).setEnabled(true);
    miClose.addActionListener(this);

    miSave = new JMenuItem ("Save");
    miSave.setAccelerator(KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_S,
java.awt.Event.META_MASK));
    fileMenu.add(miSave).setEnabled(false);
    miSave.addActionListener(this);

    miSaveAs = new JMenuItem ("Save As...");
    fileMenu.add(miSaveAs).setEnabled(false);
    miSaveAs.addActionListener(this);
```

-continued-

```
        mainMenuBar.add(fileMenu);
    }

    //-------------------------------------------
    //
    //  Create the "Edit" menu
    //
    //-------------------------------------------
    public void addEditMenuItems() {
        miUndo = new JMenuItem("Undo");
        miUndo.setAccelerator(KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_Z,
java.awt.Event.META_MASK));
        editMenu.add(miUndo).setEnabled(false);
        miUndo.addActionListener(this);
        editMenu.addSeparator();

        miCut = new JMenuItem("Cut");
        miCut.setAccelerator(KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_X,
java.awt.Event.META_MASK));
        editMenu.add(miCut).setEnabled(false);
        miCut.addActionListener(this);

        miCopy = new JMenuItem("Copy");
        miCopy.setAccelerator(KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_C,
java.awt.Event.META_MASK));
        editMenu.add(miCopy).setEnabled(false);
        miCopy.addActionListener(this);

        miPaste = new JMenuItem("Paste");
        miPaste.setAccelerator(KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_V,
java.awt.Event.META_MASK));
        editMenu.add(miPaste).setEnabled(false);
        miPaste.addActionListener(this);

        miClear = new JMenuItem("Clear");
```

-continued-

```
      editMenu.add(miClear).setEnabled(false);
      miClear.addActionListener(this);
      editMenu.addSeparator();

      miSelectAll = new JMenuItem("Select All");
      miSelectAll.setAccelerator(KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_A,
java.awt.Event.META_MASK));
      editMenu.add(miSelectAll).setEnabled(false);
      miSelectAll.addActionListener(this);
      editMenu.addSeparator();

      miParameters = new JMenuItem("Parameters...");
      editMenu.add(miParameters).setEnabled(true);
      miParameters.addActionListener(this);

      mainMenuBar.add(editMenu);
   }

   //------------------------------------------
   //
   //  Add menus
   //
   //------------------------------------------
   public void addMenus() {
      addFileMenuItems();
      addEditMenuItems();
      setJMenuBar (mainMenuBar);
   }

   //------------------------------------------
   //
   //  Main function of the class.
   //  Makes the calls to draw menus and main window.
   //
   //------------------------------------------
```

-continued-

```
public JBottomRemove() {
    super("JBottomRemove");
    this.getContentPane().setLayout(new GridBagLayout());
    addMenus();
    setSize(250,300);

    //aboutBox = new AboutBox();
    //Toolkit.getDefaultToolkit();
    //MRJApplicationUtils.registerAboutHandler(this);
    //MRJApplicationUtils.registerQuitHandler(this);

    // Initialize parameters
    maxPingGap = 150;
    windowWidth = 0.3;
    critVal = 2;
    averageWindow = 500;
    threshold = 0;
    softEdge = true;
    pingFraction = .5;

    // Create the filter button
    filterButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            doFilter();
        }
    });

    // Set layout to draw filter button
    GridBagConstraints c=new GridBagConstraints();
    c.insets=new Insets(16,16,16,16);
    c.weighty=1.0;
    c.weightx=1.0;
    c.gridx=1;
    c.gridy=1;
    c.anchor=GridBagConstraints.SOUTH;
```

-continued-

```
    this.getContentPane().add(filterButton,c);

    setVisible(true);
  }

  public void paint(Graphics g) {
                        super.paint(g);
    g.setColor(Color.blue);
    g.setFont (font);
    g.drawString(message, 40, 80);
  }

  /*public void handleAbout() {
    //aboutBox.setResizable(false);
    //aboutBox.setVisible(true);
    //aboutBox.show();
  }

  public void handleQuit() {
    System.exit(0);
  }*/

  // ActionListener interface (for menus)
  public void actionPerformed(ActionEvent newEvent) {
    if (newEvent.getActionCommand().equals(miNew.getActionCommand())) doNew();
    else if (newEvent.getActionCommand().equals(miOpen.getActionCommand())) doOpen();
    else if (newEvent.getActionCommand().equals(miClose.getActionCommand())) doClose();
    else if (newEvent.getActionCommand().equals(miSave.getActionCommand())) doSave();
    else if (newEvent.getActionCommand().equals(miSaveAs.getActionCommand())) doSaveAs();
    else if (newEvent.getActionCommand().equals(miUndo.getActionCommand())) doUndo();
    else if (newEvent.getActionCommand().equals(miCut.getActionCommand())) doCut();
    else if (newEvent.getActionCommand().equals(miCopy.getActionCommand())) doCopy();
    else if (newEvent.getActionCommand().equals(miPaste.getActionCommand())) doPaste();
    else if (newEvent.getActionCommand().equals(miClear.getActionCommand())) doClear();
    else if (newEvent.getActionCommand().equals(miSelectAll.getActionCommand())) doSelectAll();
    else if (newEvent.getActionCommand().equals(miParameters.getActionCommand())) doSetParameters();
```

-continued-

```
}

//-------------------------------------------
//
//  Methods initiated by menu selections
//
//-------------------------------------------
public void doNew() {}

public void doOpen() {
   JFileChooser chooser = new JFileChooser();
   chooser.setMultiSelectionEnabled(true);

   // Get the old file filter and remove it
   FileFilter ft = chooser.getAcceptAllFileFilter();
   chooser.removeChoosableFileFilter(ft);

   //Create and add the new file filter
   SimpleFilter myFilter = new SimpleFilter("raw","Raw Echoes File");
   //chooser.setCurrentDirectory(new File("."));
   chooser.addChoosableFileFilter(myFilter);

   // Put the open dialog up
   if (chooser.showOpenDialog(JBottomRemove.this)!=JFileChooser.APPROVE_OPTION)
      return;

   file = chooser.getSelectedFiles();

   return;
}

public void doClose() {}

public void doSave() {}

public void doSaveAs() {}
```

-continued-

```
public void doUndo() {}

public void doCut() {}

public void doCopy() {}

public void doPaste() {}

public void doClear() {}

public void doSelectAll() {}

public void doSetParameters() {
    // Create new dialog window.
    ParameterDialog parameters = new ParameterDialog(this);
    // Set the intial values.
    parameters.setMaxPingGap(maxPingGap);
    parameters.setWindowWidth(windowWidth);
    parameters.setCritValue(critVal);
    parameters.setAverageWindow(averageWindow);
    parameters.setThreshold(threshold);
    parameters.setSoftEdge(softEdge);
    parameters.setPingFraction(pingFraction);

    parameters.show();    // Display the dialog window.

    // Upon clicking "OK", get the values from the dialog box.
    if (parameters.getOption() == JOptionPane.OK_OPTION){
        maxPingGap = parameters.getMaxPingGap();
        windowWidth = parameters.getWindowWidth();
        critVal = parameters.getCritValue();
        averageWindow = parameters.getAverageWindow();
        threshold = parameters.getThreshold();
        softEdge = parameters.getSoftEdge();
        pingFraction = parameters.getPingFraction();
```

-continued-

```
            if(pingFraction>1){
                pingFraction=(double)1;
            }
        }
    }

    //-------------------------------------------
    //
    //  Method that calls class that filters data.
    //  Initiated by clicking the "Filter" button.
    //
    //-------------------------------------------
    public void doFilter(){
        // Call class that opens and filters data.
        // Should bring up a progress bar showing
        // percentage of files processed.

        BottomFilter bFilter = new
BottomFilter(this,file,maxPingGap,windowWidth,critVal,averageWindow,threshold,softEdge,pingFraction);
    }



    public static void main(String args[]) {
        new JBottomRemove();
    }

}


//
//      File:   AboutBox.java
//

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
```

-continued-

```
public class AboutBox extends JFrame
                implements ActionListener
{
   protected JButton okButton;
   protected JLabel aboutText;

   public AboutBox() {
         super();
      this.getContentPane().setLayout(new BorderLayout(15, 15));
      this.setFont(new Font ("SansSerif", Font.BOLD, 14));

      String aboutText="JBottomRemove\n"+"By Carl T. Pfisterer\n"+"Alaska Department of Fish and
Game\n"+"2002";
      JTextArea txt=new JTextArea(aboutText);
      //txt.setBorder(new EmptyBorder(5,10,5,10));
      txt.setFont(new Font("Helvetica",Font.BOLD,12));
      txt.setEditable(false);
      txt.setBackground(getBackground());
      JPanel textPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 15, 15));
      textPanel.add(txt);
      this.getContentPane().add (textPanel, BorderLayout.CENTER);

      okButton = new JButton("OK");
      JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 15, 15));
      buttonPanel.add (okButton);
      okButton.addActionListener(this);
      this.getContentPane().add(buttonPanel, BorderLayout.SOUTH);
      this.pack();
   }

   public void actionPerformed(ActionEvent newEvent) {
      setVisible(false);
   }

}
```

-continued-

```
//
//  SimpleFilter.java
//  JBottomRemove
//
//  Created by Carl Pfisterer on Tue Aug 20 2002.
//  Borrowed heavily from "Swing" by Matthew Robinson and Pavel Vorobiev
//

import javax.swing.*;
import javax.swing.filechooser.*;
import java.io.File;

//-------------------------------------------
//
//  This creates a filter for file dialogs
//
//-------------------------------------------
public class SimpleFilter extends FileFilter{

    //========================
    //
    //  Declarations
    //
    //========================
    private String m_description = null;
    private String m_extension = null;

    //========================
    //
    //  Methods
    //
    //========================
    public SimpleFilter(String extension, String description){
        m_description=description;
        m_extension = "."+extension.toLowerCase();
    }
```

-continued-

```
   public String getDescription(){
      return m_description;
   }

   public boolean accept(File f){
      if(f==null)
         return false;
      if(f.isDirectory())
         return true;
      return f.getName().toLowerCase().endsWith(m_extension);
   }

}


//
//  ParameterDialog.java
//  JBottomRemove
//
//  Created by Carl Pfisterer on Tue Aug 20 2002.
//  Copyright (c) 2002 ADF&G. All rights reserved.
//

import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;
import javax.swing.event.*;

public class ParameterDialog extends JDialog
{
   //========================
   //
   //  Declarations
   //
```

-continued-

```
//========================
protected int m_option = JOptionPane.CLOSED_OPTION;
protected JLabel lbl1;
protected JLabel lbl2;
protected JLabel lbl3;
protected JLabel lbl4;
protected JLabel lbl5;
protected JLabel lbl6;
protected JTextField txt1;
protected JTextField txt2;
protected JTextField txt3;
protected JTextField txt4;
protected JTextField txt5;
protected JTextField txt6;
protected JCheckBox softEdge;
protected boolean isSoftEdge;
protected double maxPingGap;
protected double windowWidth;
protected double critValue;
protected int averageWindow;
protected double threshold;
protected double pingFraction;

//=========================
//
//  Start of methods
//
//=========================

//-------------------------------------------
//
//  Constructor of class.
//  Creates the dialog box and sets button
//  actions.
//
//-------------------------------------------
```

-continued-

```
public ParameterDialog(JFrame owner){
    super(owner,"Parameters",true);
    getContentPane().setLayout(new BoxLayout(getContentPane(),BoxLayout.Y_AXIS));
    setSize(300,225);

    lbl1 = new JLabel("Maximum Ping Gap (In %)",JLabel.LEFT);
    lbl2 = new JLabel("Window Width (In Meters)",JLabel.LEFT);
    lbl3 = new JLabel("Critical Value (No. Of SDs)",JLabel.LEFT);
    lbl4 = new JLabel("Size of Averaging Window",JLabel.LEFT);
    lbl5 = new JLabel("Minimum Voltage Threshold (V)",JLabel.LEFT);
    lbl6 = new JLabel("Ping Fraction (decimal %)",JLabel.LEFT);
    txt1 = new JTextField(5);
    txt2 = new JTextField(5);
    txt3 = new JTextField(5);
    txt4 = new JTextField(5);
    txt5 = new JTextField(5);
    txt6 = new JTextField(5);
    softEdge =  new JCheckBox("Use Soft Border     ");
    isSoftEdge = true;
    softEdge.setSelected(isSoftEdge);

    JPanel p = new JPanel();
    p.setLayout(new FlowLayout());
    p.add(lbl1);
    p.add(txt1);
    p.add(lbl2);
    p.add(txt2);
    p.add(lbl3);
    p.add(txt3);
    p.add(lbl4);
    p.add(txt4);
    p.add(lbl5);
    p.add(txt5);
    p.add(softEdge);
    p.add(lbl6);
    p.add(txt6);
```

-continued-

```
    getContentPane().add(p);

    JPanel p1 = new JPanel(new GridLayout(1,2,20,2));
    JButton btOK = new JButton("OK");
    ActionListener lst = new ActionListener(){
        public void actionPerformed(ActionEvent e){
            m_option = JOptionPane.OK_OPTION;
            setVisible(false);
        }
    };
    btOK.addActionListener(lst);
    p1.add(btOK);

    JButton btCancel = new JButton("Cancel");
    lst = new ActionListener(){
        public void actionPerformed(ActionEvent e){
            m_option = JOptionPane.CANCEL_OPTION;
            setVisible(false);
        }
    };
    btCancel.addActionListener(lst);
    p1.add(btCancel);
    p.add(p1);
    getContentPane().add(p);
}

//-------------------------------------------
//
//  Accessor functions to SET variables
//
//-------------------------------------------
public void setMaxPingGap(double number){
    maxPingGap=number;
    txt1.setText(Double.toString(maxPingGap));
}
```

-continued-

```
public  void setWindowWidth(double number){
   windowWidth=number;
   txt2.setText(Double.toString(windowWidth));
}

public void setCritValue(double number){
   critValue=number;
   txt3.setText(Double.toString(critValue));
}

public void setAverageWindow(int number){
   averageWindow=number;
   txt4.setText(Integer.toString(averageWindow));
}

public void setThreshold(double number){
   threshold = number;
   txt5.setText(Double.toString(threshold));
}

public void setSoftEdge(boolean value){
   isSoftEdge = value;
   softEdge.setSelected(isSoftEdge);
}

public void setPingFraction(double value){
   pingFraction = value;
   txt6.setText(Double.toString(pingFraction));
}

//-------------------------------------------
//
//  Accessor functions to GET variables
//
//-------------------------------------------
public int getOption() {
```

-continued-

```
    return m_option;
  }

  public double getMaxPingGap(){
    maxPingGap=Double.parseDouble(txt1.getText());
    return maxPingGap;
  }

  public double getWindowWidth(){
    windowWidth=Double.parseDouble(txt2.getText());
    return windowWidth;
  }

  public double getCritValue(){
    critValue=Double.parseDouble(txt3.getText());
    return critValue;
  }

  public int getAverageWindow(){
    averageWindow=Integer.parseInt(txt4.getText());
    return averageWindow;
  }

  public double getThreshold(){
    threshold = Double.parseDouble(txt5.getText());
    return threshold;
  }

  public boolean getSoftEdge(){
    isSoftEdge = softEdge.isSelected();
    return isSoftEdge;
  }

  public double getPingFraction(){
    pingFraction=Double.parseDouble(txt6.getText());
    return pingFraction;
```

-continued-

```
  }
}



//
//  BottomFilter.java
//  JBottomRemove
//
//  Created by Carl Pfisterer on Wed Aug 21 2002.
//  Copyright (c) 2002 ADF&G. All rights reserved.
//

import java.awt.*;
import java.awt.event.*;
import java.util.*;
import java.lang.Math.*;
import javax.swing.*;
import javax.swing.event.*;

import java.io.File;
import java.io.*;
import java.util.Hashtable;

//public class BottomFilter extends JDialog
public class BottomFilter extends JFrame
{
  //=========================
  //
  // Declarations
  //
  //=========================
  protected JLabel dialogText;
  protected int m_option = JOptionPane.CLOSED_OPTION;
  protected int counter=0;
  protected int i=0;
  protected double percentDone;
```

-continued-

```
protected JProgressBar progress;
protected int pMin=0;                  //Minimum value of progress bar
protected int pMax=100;                         //Maximum value of progress bar



protected double windowWidth;
protected double maxPingGap;
protected double critValue;
protected double missedPings;
protected int averageWindow;
protected double threshold;
protected boolean softEdge;
protected double pingFraction;

//=========================
//
// Methods
//
//=========================

//----------------------------------------
//
// Constructor.  This method brings up a window
// showing a progress bar and the current file
// being processed.  When the filtering of all
// the files is finished, the "OK" button is enabled.
//
//----------------------------------------
public BottomFilter(JFrame owner,File[] files,double m_PingGap,double w_Width,double c_Value,int a_Window, double thresh,boolean s_Edge,double p_Fraction){
    super("Filtering");

    getContentPane().setLayout(new BoxLayout(getContentPane(),BoxLayout.Y_AXIS));
    setSize(235,120);

    windowWidth = w_Width;
```

-continued-

```
maxPingGap = m_PingGap;
critValue = c_Value;
missedPings = m_PingGap;
averageWindow = a_Window;
threshold = thresh;
softEdge = s_Edge;
pingFraction=p_Fraction;

UIManager.put("ProgressBar.selectionBackground",Color.black);
UIManager.put("ProgressBar.selectionForeground",Color.white);
UIManager.put("ProgressBar.foreground",new Color(8,32,128));

dialogText = new JLabel("Processing: File Name Here",JLabel.LEFT);
progress = new JProgressBar();
progress.setMinimum(pMin);
progress.setMaximum(pMax);
progress.setStringPainted(true);

JButton btOK = new JButton("OK");
ActionListener lst = new ActionListener(){
   public void actionPerformed(ActionEvent e){
      m_option = JOptionPane.OK_OPTION;
      setVisible(false);
   }
};

JPanel p = new JPanel();
p.setLayout(new FlowLayout());

btOK.addActionListener(lst);
btOK.setEnabled(false);                    //Set to false until filtering completed.

p.add(dialogText);
p.add(progress);
p.add(btOK);
getContentPane().add(p);
```

-continued-

```
    setVisible(true);

    final File[] allFiles=files;
    final int numFiles = allFiles.length;
    /*Thread runner = new Thread(){
        public void run(){
            counter=0;
            for(i=0;i<numFiles;i++){
                Runnable runme = new Runnable(){
                    public void run(){
                        progress.setValue((int)(((double)i/(double)(numFiles))*100));
                        dialogText.setText("Processing:"+allFiles[i].getName());
                        processFile(allFiles[i]);
                    }
                };
                SwingUtilities.invokeLater(runme);
                counter++;
                try{
                    Thread.sleep(100);
                }
                catch (Exception ex){}
            }
        }
    };
    runner.start();*/
    for(i=0;i<numFiles;i++){
        progress.setValue((int)(((double)i/(double)(numFiles-1))*100));
        dialogText.setText("Processing:"+allFiles[i].getName());
        processFile(allFiles[i]);
    }

    btOK.setEnabled(true);          //Now that filtering is finished, enable the OK button.
}

//----------------------------------------------
```

-continued-

```
//
// Method that begins processing file
//
//---------------------------------------------
protected void processFile(File file){
    RangeData[] rangeData;

    double maxRange=getMaxRange(file);
    rangeData = new RangeData[(int)(maxRange/windowWidth+2)];  //Add 2 as a precaution
    for(int j=0;j<rangeData.length;j++){
        rangeData[j] = new RangeData(averageWindow);
    }
    filterData(file,rangeData);
}

protected double getMaxRange(File file){
    //Skip the header rows, then loop through the data to get
    //the maximum range from all the sequences.
    boolean done = false;
    String buffer = new String();
    File outFile = file;
    double max=0;

    try{
        BufferedReader in = new BufferedReader(new FileReader(file));
        buffer = in.readLine();
        while(!done){                    //First skip over the header rows
            if(buffer.startsWith("* Start")){
                done = true;
            }
            else{
                buffer = in.readLine();
            }
        }
        done = false;
        buffer = in.readLine();
```

-continued-

```
        while(!done){
          if(buffer.startsWith("* Start")||buffer.startsWith("* End")){
            buffer = in.readLine();
          }
          else{
            if(buffer.startsWith("* Stop Processing")){
              done = true;
            }
            else{
              double range = extractNumber(buffer.toCharArray(),1);
              if(range>max){
                max=range;
              }
              buffer = in.readLine();
            }
          }
        }
      }
      catch(Exception e){
        e.printStackTrace();
      }

      return max;
    }

    protected void filterData(File file,RangeData[] data){
      boolean done=false;
      String buffer = new String();
      File outFile;
      double n=0;

      try{
        outFile = new File((file.getPath().substring(0,file.getPath().length()-4)).concat("f.raw"));
        BufferedReader in = new BufferedReader(new FileReader(file));
        BufferedWriter out = new BufferedWriter(new FileWriter(outFile));
```

-continued-

```
// First read and output the header rows.
buffer = in.readLine();
while(!done){
    if(buffer.startsWith("* Start")){
        done = true;
        out.write(buffer,0,buffer.length());
        out.newLine();
    }
    else{
        out.write(buffer,0,buffer.length());
        out.newLine();
        buffer = in.readLine(); //Increment to next line
    }
}
done=false;         //reset done value

//Now continue and start calculating averages and writting rows meeting criteria.
buffer = in.readLine();
double lastPing=1;
while(!done){
    if(buffer.startsWith("* End")){
        //clear data
        for(int i=0;i<data.length;i++){
            if(data[i]!=null){
                data[i].clear();
            }
        }
        out.write(buffer,0,buffer.length());
        out.newLine();
        buffer = in.readLine();
    }
    else{
        if(buffer.startsWith("* Start")){
            out.write(buffer,0,buffer.length());
            out.newLine();
            buffer = in.readLine();
```

-continued-

```
            }
            else{
              if(buffer.startsWith("* Stop Processing")){        //Modify for EOF condition
                out.write(buffer,0,buffer.length());
                out.newLine();
                done=true;
              }
              else{
                double voltage=extractNumber(buffer.toCharArray(),2);
                double range=extractNumber(buffer.toCharArray(),1);
                double pingNumber=extractNumber(buffer.toCharArray(),0);
                data[(int)(range/windowWidth)].addValue(voltage,(int)pingNumber,(double)1);
                if((int)(range/windowWidth)>0 && softEdge == true){
                  data[(int)(range/windowWidth)-1].addValue(voltage,(int)pingNumber,pingFraction);
                }
                if((int)(range/windowWidth)<data.length && softEdge == true){
                  data[(int)(range/windowWidth)+1].addValue(voltage,(int)pingNumber,pingFraction);
                }
                lastPing=pingNumber;

                if(softEdge==false){
                  n=(double)data[(int)(range/windowWidth)].getn();
                }
                else{
                  n=data[(int)(range/windowWidth)].getAdjN();
                }

if(((voltage<data[(int)(range/windowWidth)].getMean()+critValue*data[(int)(range/windowWidth)].getsd())&&(dat
a[(int)(range/windowWidth)].getMissedPings(softEdge)<missedPings)&&(n>50))||(voltage<threshold)){
                  boolean include = false;
                }
                else{
                  out.write(buffer,0,buffer.length());
                  out.newLine();
                }
                buffer = in.readLine();
```

-continued-

```
                }
            }
        }
    }
    in.close();
    out.close();
  }
  catch(Exception e){
    e.printStackTrace();
  }
}

//=====================================
//Function extracts a number from a string that contains many groups
//of numbers or characters seperated by spaces.  Receives a string and
//the number of groups of characters or numbers to skip and returns
//the number of type float.
//=====================================
public double extractNumber(char dataStr[],int numSkip)
{
  char[] numStr=new char[10];
  int flag=0;
  int cntr=1,cntr2;
  boolean done=false;

  while(!done)
  {
      cntr2=0;
      if(dataStr[cntr]!=' ')
      {
          while(dataStr[cntr+cntr2]!=' ')
          {
              if(flag==numSkip)       //how many groups of numbers to skip
              {
                  numStr[cntr2]=dataStr[cntr+cntr2];
                  numStr[cntr2+1]='\0';
```

-continued-

```
                    done=true;
                }
                cntr2++;
            }
            flag++;
        }
        if(cntr2>0)
            cntr=cntr+cntr2;
        else
            cntr++;
    }
    String tempString=new String(numStr);
    return (Double.valueOf(tempString)).doubleValue();
  }

}


//
//  RangeData.java
//  JBottomRemove
//
//  Created by Carl Pfisterer on Fri Aug 23 2002.
//  Copyright (c) 2002 ADF&G. All rights reserved.
//

import java.util.*;
import java.lang.*;
import java.lang.Math.*;


public class RangeData {

  //==============================
  // Declarations
  //==============================
```

-continued-

```
protected double[] values;
protected int[] pingNumbers;
protected double[] weight;
protected double mean;
protected double sd;
protected int n;
protected int width;

//================================
// Methods
//================================

//---------------------------------------------
//
// Constructor Method
//
//---------------------------------------------
public RangeData(int avgWindow){
   values = new double[avgWindow];
   pingNumbers = new int[avgWindow];
   weight = new double[avgWindow];
   n=0;            //No values entered yet
   width=avgWindow;
}

public RangeData(){
   int avgWindow=200;
   values = new double[avgWindow];
   pingNumbers = new int[avgWindow];
   n=0;
   width=avgWindow;
}
//---------------------------------------------
//
// Takes an array of doubles and returns the mean
//
```

-continued-

```
//----------------------------------------------
protected void mean(){
    double sum=0,adjN=0;

    for(int i=0;i<n;i++){  //Calculating the weighted average
        sum=sum+weight[i]*values[i];
        adjN=adjN+weight[i];
    }
    mean = (sum/adjN);
}

//----------------------------------------------
//
// Takes an array of doubles and returns the
// standard deviation.
//
//----------------------------------------------
protected double sd(){

    double sse=0;

    // First loop through the data and add the squared errors
    for(int i=0;i<n;i++){
        sse=sse+Math.pow((mean-values[i]),(double)2);
    }

    sse=sse/(n-1);            // Then divide by n-1 giving the variance
    sse=Math.pow(sse,.5);// Then take the square root giving the standard deviation

    sd = sse;
    if(n<2){
        sd=0;
    }
    return sd;
}
```

-continued-

```
//---------------------------------------------
//
// Adds a new number to the array of values.
// If the end of the array has been reached, it
// drops the oldest value, shifts the array and
// adds the new value to the end.
//
//---------------------------------------------
public void addValue(double number,int pingNum,double fraction){

    if(n<(width)){
        n++;
        values[n-1]=number;
        pingNumbers[n-1]=pingNum;
        weight[n-1]=fraction;
        mean();
    }
    else{
        for(int i=0;i<n-1;i++){
            values[i]=values[i+1];
            pingNumbers[i]=pingNumbers[i+1];
            weight[i]=weight[i+1];
        }
        values[n-1]=number;
        pingNumbers[n-1]=pingNum;
        weight[n-1]=fraction;
        mean();
    }
    //Note: it should be quicker to calculate the mean here rather than each time
    //calculating the sse for the standard deviation.
}

//---------------------------------------------
//
// Returns the percentage of missed pings
// within the dataset.
```

-continued-

```
//
//---------------------------------------------
protected double missedPings(){
   return 100*(1-(n/(pingNumbers[n-1]-pingNumbers[0])));
}

protected double adjMissedPings(){
   double adjN=0;

   for(int i=0;i<n;i++){  //Calculating the adjusted n
      adjN=adjN+weight[i];
   }
   return 100*(1-(adjN/(pingNumbers[n-1]-pingNumbers[0])));
}

//---------------------------------------------
//
// Accessor functions to get the mean and
// standard deviation values.
//
//---------------------------------------------
public double getMean(){
   return mean;
}

public double getsd(){
   return sd();
}

public int getn(){
   return n;
}

public double getAdjN(){
   double adjN=0;
```

-continued-

```
        for(int i=0;i<n;i++){
            adjN=adjN+weight[i];
        }
        return adjN;
    }

    public double getMissedPings(boolean softEdge){
        if(softEdge){
            return adjMissedPings();
        }
        else{
            return missedPings();
        }
    }

    public void clear(){
        n=0;
    }

}
```

APPENDIX C. CLIMATOLOGICAL AND HYDROLOGIC OBSERVATIONS

Appendix C1.–Climatological and hydrologic observations at the Sheenjek River project site, 2004.

				Wind	Temperature (C°)			Water Level (cm)			
			Cloud			Air				Water	
		Precipitation	Cover	Direction and	Water			± 24 h	relative to	Color	
Date	Time	(code)[a]	(code)[b]	velocity (mph)	Surface	Minimum	Maximum	Change	zero datum	(code)[c]	Remarks
08-Aug	1900	A	S	SW 4	16.6	27	28	zero datum	0.0	A	Wizard installed 1600: Smoke & clouds.
09-Aug	1900	B	S	SW 4	16.8	10	31	-5.0	-5.0	A	Smoke & haze in am: hot & sunny.
10-Aug	1900	B	C	calm	17.0	10	31	-5.0	-10.0	A	Hot & sunny all day; evening clouds.
11-Aug	1900	B	C	SW 2	17.2	13	30	-3.0	-13.0	A	Hazy, hot & sunny, smoke after 2000.
12-Aug	1900	A	S	S6	16.8	10	27	-2.0	-15.0	A	Smoky to 1500; hazy sun in pm.
13-Aug	1900	B	O	S 3	15.8	12	27	0.0	-15.0	A	Overcast; very light drizzle 1700-2000.
14-Aug	1900	A	S	SE 3	16.0	12	26	-2.0	-17.0	A	Partly cloudy, nice day; wind max 25.
15-Aug	1900	A	C	calm	16.6	N/A	29	-1.0	-18.0	A	Sunny & hot; no min temp.
16-Aug	1900	A	C	E 2	16.6	7	30	-1.0	-19.0	A	Sunny & hot; wind max 12 at 0046.
17-Aug	1900	A	C	calm	16.6	9	30	4.0	-15.0	A	Cloudy am, hazy sunny hot pm.
18-Aug	1900	A	F	calm	16.8	N/A	30	1.0	-14.0	A	Smoky all day, sunny hot afternoon.
19-Aug	1900	A	F	N-NE 5-19	16.8	13	29	-1.0	-15.0	A	Smoky all day, increased in pm.
20-Aug	1900	A	S	N 5-23	17.0	15	28	-1.0	-16.0	A	Windy all day, sunny and clear day.
21-Aug	1900	A	C	NE 5-17	16.1	11	25	0.0	-16.0	A	Breezy all day, sunny and warm.
22-Aug	1900	A	C	NE 0-14	15.4	10	26	2.0	-14.0	A	Sunny with breeze, cool am, warm pm.
23-Aug	1900	A	S	NE 5-27	14.4	6	23	0.0	-14.0	A	Strong winds all day, smoky in am.
24-Aug	1900	A	S	N 5-26	12.8	5	16	-1.0	-15.0	A	Strong winds all day, cool & sunny.
25-Aug	1900	A	B	N-NE10-22	11.0	2	16	-2.0	-17.0	A	Windy, overcast & cool; wind max 22.
26-Aug	1900	A	B	N-NE 14	10.0	2	14	-1.0	-18.0	A	Extremely windy, sunny & chilly.
27-Aug	1900	A	B	N-NE 4	10.2	2	17	1.0	-17.0	A	Windy, sunny all day, clouds 18:30.
28-Aug	1900	A	S	SW-0	10.0	1	19	-3.0	-20.0	A	Sunny, real smoky am & late pm.
29-Aug	1900	A	F	E-0	9.0	-3	12	-2.0	-22.0	A	Dense smoke all day, calm winds.
30-Aug	1900	B	F	NE 6-8	8.4	4	13	-2.0	-24.0	A	Dense smoke all day, wind in pm.
31-Aug	1900	A	B	SW -3	9.8	7	20	-2.0	-26.0	A	Smoke cleared, high clouds.
01-Sep	1900	B	B	SW-4	9.8	2	18	-0.5	-26.5	A	Smoke in AM, intermittent rain to 19:00.
02-Sep	1900	A	S	SW-2	10.0	2	19	-0.5	-27.0	A	No smoke, good fall day!
03-Sep	1900	A	S	SW 4	9.4	2	17	-2.0	-29.0	A	Gorgeous sunny fall day.
04-Sep	1900	B	B	S 8	8.2	-6	16	-2.0	-31.0	A	Cold night, broken overcast.
05-Sep	1900	B	B	N-NE 4	8.0	3	18	-1.0	-32.0	A	Cold, mixed snow/rain, partly cloudy.
06-Sep	1900	A	C	SW 1	7.6	-8	13	-1.0	-33.0	A	Cold night!!! Sunny cool fall day.
07-Sep	1900	A	S	0-calm	7.4	-6	13	-1.0	-34.0	A	Cold night!! Thin cirrus all day, calm.
08-Sep	1900	A	C	0-calm	7.0	-7	15	-2.0	-36.0	A	Clear sunny day; smoke haze to SW.
09-Sep	1900	A	C	0-calm	6.4	-7	12	-2.0	-38.0	A	Cold night; clear sky thin smoke haze.

-continued-

Appendix C1.–Page 2 of 2.

				Wind	Temperature (C°)			Water Level (cm)			
			Cloud							Water	
		Precipitation	Cover	Direction and	Water	Air		± 24 h	relative to	Color	
Date	Time	(code)[a]	(code)[b]	velocity (mph)	Surface	Minimum	Maximum	Change	Zero datum	(code)[c]	Remarks
10-Sep	1900	A	C	0-calm	5.1	-5	13	-2.0	-40.0	A	Sunny fall day; light smoke haze.
11-Sep	1900	A	S	3-N	5.8	2	10	-1.0	-41.0	A	Sunny, day, scattered clouds.
12-Sep	1900	A	S	2-N	6.0	-9	11	-2.0	-43.0	A	New water gauge=36, old gauge=1.
13-Sep	1900	A	O	8-N	6.2	-1	12	-1.0	-44.0	A	Scattered clouds most of day.
14-Sep	1900	A	B	12- N	5.2	1	6	-2.0	-46.0	A	Mostly cloudy, windy, cold day.
15-Sep	1900	F	O	5-N	4.1	-4	4	-1.0	-47.0	A	Overcast, cold, windy, snow flurries.
16-Sep	1900	F	O	9-N	3.2	-8	5	-1.0	-48.0	A	Overcast, cold, windy, snow flurries.
17-Sep	1900	A	C	4-N	3.4	-4	6	-1.5	-49.5	A	Clear, sunny day, winds in am only.
18-Sep	1900	A	C	4-N	3.2	-11	6	-0.5	-50.0	A	Clear skies, sunny all day.
19-Sep	1900	A	C	N 2	3.4	-5	7	-2.0	-52.0	A	Sunny & mostly clear skies.
20-Sep	1900	A	O	N 6	4.8	-1	11	-1.0	-53.0	A	Partly cloudy am, warmer somewhat.
21-Sep	1900	C	C	S 5	4.8	2	4	-0.5	-53.5	A	High overcast, continuous light rain.
22-Sep	1900	A	S	NE-2	5.0	-2	9	-3.5	-57.0	A	Scattered clouds & light winds.
23-Sep	1900	G	O	SW-14-18	3.6	-1	9	-0.5	-57.5	A	15:15 started to rain turned to snow.
24-Sep	1900	A	S	SE-2	3.2	-8	5	-0.5	-58.0	A	Scattered clouds w/light winds.
25-Sep	1900	A	B	Calm	3.0	-3	8	-1.0	-59.0	A	Scattered clouds, snow melted.
	Average				9.9	2	17				

[a] Precipitation code for the preceding 24-hr period: A = None; B = Intermittent rain; C = Continuous rain; D = snow and rain mixed; E = light snowfall; F = Continuous snowfall; G = Thunderstorm w/ or w/o precipitation.

[b] Instantaneous cloud cover code: C = Clear and visibility unlimited (CAVU); S = Scattered (<60%); B = Broken (60-90%); O = Overcast (100%); F = Fog or thick haze or smoke.

[c] Instantaneous water color code: A = Clear; B = Slightly murky or glacial; C = Moderately murky or glacial; D = Heavily murky or glacial; E = Brown, tannic acid stain.

APPENDIX D. AGE COMPOSITION ESTIMATES

Appendix D1.–Age composition estimates of Sheenjek River fall chum salmon, 1974–2004.

Year[a]	Sample (readable)	Age-3	Age-4	Age-5	Age-6	Estimated Escapement
1974[b]	136	0.669	0.301	0.029	0.000	89,966
1975[b]	197	0.036	0.949	0.015	0.000	173,371
1976[b]	118	0.017	0.441	0.542	0.000	26,354
1977[b]	178	0.112	0.725	0.163	0.000	45,544
1978[b]	190	0.079	0.821	0.100	0.000	32,449
1979	none					91,372
1980	none					28,933
1981[c]	340	0.029	0.850	0.118	0.003	74,560
1982[c]	109	0.030	0.470	0.490	0.010	31,421
1983[c]	108	0.065	0.870	0.065	0.000	49,392
1984[d]	297	0.101	0.805	0.094	0.000	27,130
1985[d]	508	0.012	0.927	0.061	0.000	152,768
1986[d]	442	0.081	0.412	0.500	0.007	84,207
1987[d]	431	0.021	0.898	0.072	0.009	153,267
1988[d,e]	120	0.025	0.683	0.292	0.000	45,206
1989[d,e]	154	0.052	0.766	0.169	0.013	99,116
1990[d]	143	0.028	0.706	0.252	0.014	77,750
1991[d]	147	0.000	0.592	0.395	0.014	86,496
1992[d]	134	0.000	0.179	0.806	0.015	78,808
1993[d,e]	192	0.005	0.640	0.339	0.016	42,922
1994[d]	173	0.012	0.561	0.405	0.023	153,000
1995[d]	166	0.012	0.542	0.386	0.060	235,000
1996[d]	191	0.016	0.330	0.618	0.037	248,000
1997	none					80,423
1998	only 3 fish					33,058
1999	none					14,229
2000	none					30,084
2001[f]	71	0.000	0.352	0.648	0.000	53,932
2002[g]	31	0.000	0.613	0.387	0.000	31,642
2003[d]	84	0.012	0.821	0.155	0.012	44,047
2004[d]	104	0.115	0.615	0.250	0.019	37,878
Avg 1974-03		0.059	0.636	0.296	0.010	80,482
Avg 1974-85		0.115	0.716	0.168	0.001	68,605
Avg 1986-03		0.019	0.578	0.387	0.016	88,399
Even Years		0.088	0.527	0.376	0.009	77,161
Odd years		0.030	0.744	0.215	0.011	93,096

[a] Age determination from scales for years 1974–1985; and from vertebrae 1986–2004.
[b] Carcass samples from spawning grounds.
[c] Escapement samples taken with 5-7/8 inch gillnets at rkm 10.
[d] Escapement samples taken with beach seine rkm 5–20.
[e] Escapement samples were predominantly taken late in run.
[f] 68 carcass samples and 5 beach seine samples collected between rkm 11 and 25.
[g] 28 beach seine samples collected at rkm 13 and 1 carcass collected at rkm 10.